ANDROID APPLICATION TO SEARCH FOR CHURCH LOCATIONS IN YOGYAKARTA

Tugas Akhir

Diajukan untuk Memenuhi Salah Satu Persyaratan Mencapai Derajat Sarjana Teknik Informatika

Dibuat Oleh:

ALBERTUS ALVIN ADHITAMA SUSANTO
160708805

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ATMA JAYA YOGYAKARTA

2020

To: Dean Xiangjun Zhao
School of Information Engineering

This thesis, written by Albertus Alvin Adhitama Susanto, and entitled Android Application to Search for Church Locations in Yogyakarta, having been approved in respect to style and intellectual content, is referred to you for judgment.

We have read this thesis and recommend that it be approved.

Qing Li

Haiyong Wu

Wan Li Song

Date of Defense: May 19, 2020

The thesis of Albertus Alvin Adhitama Susanto is approved.

Dean Xiangjun Zhao
School of Information Engineering

Zhu Minghui
Chief of Foreign Affairs Office

Nanjing Xiaozhuang University, 2020

ACKNOWLEDGEMENTS

Assalamualaikum Wr. Wb., Salam Damai Sejahtera, Shalom, Om Swastiastu, Namo Buddhaya, Salam Kebajikan, Salam Rahayu. Praise and thank God Almighty, who gave His gifts and blessings so that the author can finish this thesis well and on time. Thesis with the title "Android Application to Search for Church Locations in Yogyakarta" was made to complete one of the requirements to get a bachelor's degree in Software Engineering Study Program, School of Information Engineering, Nanjing Xiaozhuang University and Informatics Engineering Study Program, Faculty of Industrial Technology, Universitas Atma Jaya Yogyakarta.

The author realizes that completing this thesis can't be separated from the help, guidance, support, enthusiasm, and prayers of various parties involved both directly and indirectly. On this occasion, the author would like to thank:

1. The Lord Jesus Christ for all the gifts, guidance, strength, health, and blessings so that the writer can finish the thesis well and properly.
2. Father, Mother, and my sister, who always provide prayer, guidance, advice, input, motivation, and moral also material support so that the author can finish this thesis well and on time.
3. Nizhen laoshi (倪震), as my thesis supervisor who has provided guidance, guidance, and direction to the author so that this thesis can be completed properly.
4. Mr. Martinus Maslim, S.T., M.T., as the head of the Informatics Engineering study program, which has provided the opportunity for the author to continue their education in China and has provided assistance, encouragement, and guidance.
5. The lecturer, staff, and employees at Nanjing Xiaozhuang University who have helped the author during my studies in China.
6. The lecturer, staff, and employees at the UAJY Informatics Engineering who have helped writers with their lectures at Universitas Atma Jaya Yogyakarta.
7. My entire family, who always gives encouragement and encouragement to writers during college.

8. My beloved girlfriend, Asteria Agusti Yuwita Putri, who has provided support, encouragement, assistance, prayer, and love. Thank you for always accompany me until now.
9. My friends and best friends at IT Nanjing Xiaozhuang University 2018.
10. My best friend, group kuli4h Bhre.
11. My friends at UAJY Informatics Engineering 2016.
12. All parties that cannot be mentioned one by one who has assisted directly or indirectly.

The author realizes that this thesis is far from perfect because it has limited time and knowledge of the author. Therefore, all constructive criticism and suggestions are highly expected. Finally, I hope this thesis can be useful for everyone. Wassalamualaikum Wr. Wb., Salam Damai Sejahtera, Shalom, Om Shanti Shanti Shanti Om, Sabbe Satta Bhavantu Sukhitatta, Salam Kebajikan, mugya tansah pinaringan rahayu sagung dumadi, semoga semua makhluk berbahagia.

Yogyakarta, May 7th, 2020

Albertus Alvin Adhitama Susanto (帅光者)

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

The availability of information about the Catholic church in the Special Province of Yogyakarta is still limited. The lack of available media information makes it difficult for local and out-of-town communities to get information about the location of the church where they want to worship. The solution taken to overcome this problem is to build applications that can process church data into useful information for the community. This application is an Android application based on Location-Based Service technology and utilizes the Google Maps API, Firebase, and Global Positioning System in its development. This application will make it easy for users to get information about the name of the church, church address, church contacts, worship schedules, church social media, church facilities, church photos, church maps, and the latest information.

Keywords: Catholic Church, Android, Firebase, Google Maps.

CHAPTER 1. INTRODUCTION

1.1 Background

Everyone in Indonesia has a religion. Based on the 1945 Constitution of the Republic of Indonesia, it is explained that everyone is free to embrace religion and worship according to their religion. Everyone has the right to freedom of belief. The state guarantees the independence of each population to embrace their respective religions and to worship according to their religion and beliefs. In Indonesia, there are six religions recognized by the state, namely Islam, Catholicism, Christianity, Hinduism, Buddhism, and Confucianism. The majority of Indonesians follow Islam so that the places of worship for other religions are not as much as the religion of Islam.

In 2010 it was noted that there were 6,907,873 Indonesians who were Catholic or around 2.91% of Indonesia's population (Statistics Indonesia, 2019). In the Special Province of Yogyakarta alone, there are 166,902 Catholics (Directorate General of Population and Civil Registration, 2019).

The church is a place for Catholics to worship. The Catholic Church, as a place of worship for Catholics, is an infrastructure for seeking and relating to God. In Indonesia, the search for church locations is still done manually, by looking at a map or by asking local people. This is a problem that must be solved because public services in the form of churches are quite challenging to find, especially for immigrants. Only churches in the middle of the city or people who are quite densely populated are often visited by people to worship. As for churches on the edge of the city that are not densely populated, there are only a few followers. Churches that are not located in strategic places are also rarely visited by people. This causes people only to know a few churches. Most people only come to a church with quite many followers so that there is an imbalance of facilities between churches in the city center and those outside the cities. People from outside Yogyakarta also have difficulty in finding a place to worship. They must ask the local people. People also have difficulty in obtaining information about the church, such as worship time, church location, and the latest news.

The rapid development of information technology makes it very easy for humans to run and do their daily activities. This is due to the increasing human need for practical and fast matters, with internet media that can facilitate all kinds of information retrieval, likewise, in the case of the difficulty of finding information and the location of a church.

The limited community knows the location of the church close to the current position can cause someone confusion and must ask people around the neighborhood of the church needed. It can also create a misunderstanding about the information provided by someone who was asked because there was no knowledge about the location of the church from the current location.

The migrants who are not familiar with the Yogyakarta area will feel confused to find the location of the nearest church and following their wishes. The solution to this problem is a system that can make it easy for migrants to get information about the location of the church. The right method to handle the issue is by creating a location-based service (LBS) based church search mobile application development system on the android platform, and later, it will be able to facilitate the users.

One problem that becomes an obstacle is the lack of available information and the use of information technology that is less than optimal. On the other hand, technological developments tend to lead to mobile technologies such as Android, which is currently a smartphone operating system that is quite popular among the general public. Android is a Linux-based mobile device operating system. One of the advantages of Android compared to other device operating systems is that it is relatively cheap compared to other smartphone operating systems. Besides that, both are open source so that users can develop features that do not yet exist as they wish (Safaat, 2012).

The system will be supported by a NoSQL-based realtime database owned by Firebase. Not only the database but also the authentication feature so that users can log in using email and Google accounts. The application will also use asset storage features such as images owned by Firebase Storage. The reason for using this database is because of the realtime nature required for tracking (Wiratno & Hasturi, 2017).

The utilization of an Android-based mobile application is the most suitable device for the solution to the problem above. Android is also a mobile operating system that is open source for developers to create applications. The developed application is expected to be able to display maps containing information about churches in the Special Province of Yogyakarta by utilizing Location Based Services (LBS) services and Global Positioning System (GPS) technology on Android devices. By using the application, users can find out their position and the location of the church that is nearby. The developed application will also show the closest route that users can take to get to the location of the church. This application will be able to provide realtime notifications about church information by utilizing the Firebase Cloud Messaging feature.

With the references above, an Android-based mobile application will be design and built to solve the problems faced by Catholics in Yogyakarta. An application that can display information about the location of the church location as well as displaying a brief profile, mass schedule, and current information on each church. This application can help people who come from other regions to find the church that is around them.

1.2 Formulation of the problem

Based on the existing problems in the background, the core problem that the writer can conclude is how to make an Android mobile application to search for LBS-based church locations with Firebase and Google Maps API containing church information in the form of church names, church addresses, church contacts, worship schedules, media church social, church facilities, church photos and church maps that can make it easier for Catholics to find churches in Yogyakarta.

1.3 Scope of the problem

The Church Finder mobile application created has the following limitations:

1. The application only includes churches in the Special Region of Yogyakarta.
2. The application is only available on the Android platform.
3. The application must use an internet connection.
4. The application requires GPS features on a mobile device.

1.4 Research Purposes

The purpose of this research is to build a mobile application for locating church locations to help Catholics in Yogyakarta to find church locations and find church information in the form of church names, church addresses, church contacts, worship schedules, church social media, church facilities, church photos and church maps using Firebase technology and Google Maps API.

1.5 Writing Structure

For the systematical structure, this thesis is divided into six chapters, with an explanation for each chapter is as follows:

CHAPTER 1. INTRODUCTION

This chapter contains the background, formulation of the problem, scope of the problem, research purposes, and writing structure.

CHAPTER 2. REVIEW OF LITERATURE

This chapter contains the basic theories that will be used as guidelines and references in problem-solving. This chapter also includes a literature review, which is about research that has been done before relating to the topic discussed and an explanation of the comparison between research that has been done before and the research that will be conducted.

CHAPTER 3. RESEARCH METHODOLOGY

This chapter contains the research methods that will be used. The author divides into six stages, namely literature study methods, data collection methods, analysis of software requirements, software design, software implementation, and software testing.

CHAPTER 4. SOFTWARE ANALYSIS AND DESIGN

This chapter will discuss software analysis and design, such as system analysis, product perspective, problem scope analysis, data requirements analysis, external interface requirements, product functions, functional software requirements, entity-relationship diagrams, software architecture design, and interface design.

CHAPTER 5. SOFTWARE IMPLEMENTATION AND TESTING

This chapter provides an explanation of how to implement and use the system and the results of software testing.

CHAPTER 6. SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

This chapter contains a summary, conclusions from the research discussion, and development suggestions for further research.

LIST OF REFERENCES

This section contains the bibliography used in the discussion of this thesis, both from journals, theses, books, and the internet.

CHAPTER 2. REVIEW OF LITERATURE

2.1 Basic Theory

2.1.1 Android

Android is a Linux-based operating system that is modified for mobile devices consisting of operating systems, middleware, and major applications (Juhara, 2016). Android is growing and developing in the middle of other operating systems that are also developing now, providing a different place for application developers. Android also does not distinguish the main application with third-party applications. Android users are also freed to be able to delete core applications from Android and replace them with applications from third parties (Munir et al., 2015).

Application considerations that will be developed on the Android operating system (Safaat, 2013) are as follows: (1) Complete, Android is a safe operating system and provides many tools in building software and allows for application development opportunities. (2) Open, the Android platform is provided through an open-source license. Developers can freely develop applications. (3) Free, Android is a free application platform for developers. There are no licenses or royalty fees for application development on the Android platform.

2.1.2 Location-Based Service

Location-Based Service is an information service that can be accessed using mobile devices through the internet and cellular networks that use GPS systems and aims to provide location information services or position instructions to users (Safaat, 2013). At this time, the needs of information systems must be fast, clear, and easily accessible. Moreover, the need to search for a particular place, then users can use Google Maps facilities (Rahmani et al., 2016).

LBS combines the processes of mobile services with the geographical position of users. The target position where a target is the user of location-based services themselves and other entities that are united in a function (Darma et al., 2012). LBS on mobile devices use GPS technology to take a geographical position, and then the location data will be sent to the server for processing (Segara and Subari, 2017).

On Android devices, location-based services use GPS for accuracy. Energy-intensive results and can use the network of communication service providers by utilizing signal transmitters from communication service providers with their superiority in getting results in a closed room or location, but the position results obtained are not as accurate as using GPS. However, it can save energy compared to GPS so that the combination of the two-location data collection will be very precise (Juhara, 2016).

2.1.3 Global Positioning System (GPS)

Global Positioning System is a system composed of a series of satellites that are useful for knowing the position of the location of something on the surface of the earth. The satellite sends a signal, and then the signal can be detected by a receiver on the surface of the earth. A tool to determine the position of the GPS receiver that functions to receive signals sent from GPS satellites. This GPS receiver will collect information from GPS satellites such as time, location, speed, save location, the direction of travel, and data accumulation (Anwar et al., 2015). This system is designed to provide position and speed of three dimensions as well as information to recognize time sustainably throughout the world without depending on time and weather (Pratami et al., 2014).

2.1.4 Google Maps

Google Maps is an online map that can be used easily through free services from Google. Google offers a variety of features, one of which is Google Maps. Google Maps offers satellite images for the entire world, offering a search for a place and a travel route. This service even provides an Application Programming Interface (API), which allows other developers to utilize this application in applications created (Mahdia & Noviyanto, 2013).

Google Maps is a virtual global map that can be accessed online and free of charge that has been provided by Google. In the Google Maps application, the available maps can be dragged to make it easier for users to use the app, and Google Maps also offers a search for a place to be sought and can search for the user's travel route (Mahdia & Noviyanto, 2013).

2.1.5 Google Maps API

Application Programming Interface (API) is a technology that functions to facilitate the exchange of data between two or more software. Google offers the opportunity to develop applications that have been made open source through the Application Programming Interface (API) technology. Google has a map service in the form of an API called the Google Maps API.

The Google Maps API is an application interface that can be accessed via JavaScript so that Google Maps data is displayed on certain web pages or SDKs according to the platform (Apriyani et al., 2012). Developers can customize the presentation of information from maps provided by Google for free. Application developers do not need to create a basic map that is relatively complicated because the process contains elements such as latitude, longitude, and other geospatial coordinates.

2.1.6 Firebase

Database design is a picture or diagram that shows table information that will be created, stored, and used in a database system using the form of relationships between tables in the database (Shalahudin & Rosa, 2016). The database has two types of query structures, namely SQL and NoSQL. Firebase is a database that has a NoSQL database query structure type. NoSQL databases arrange parts in other parts (subsets). Each section will have several more parts in it. This NoSQL database is suitable and is used for storing huge applications or data. By using NoSQL, data can be accessed very flexible, and there is very little chance of error when accessing a lot of data with different formats (Subastian, 2018).

Firebase is a back end as a service, which is a back-end service that is now being developed by Google. Firebase is developed on Google's infrastructure and is scaled automatically, even for the most extensive applications. Firebase itself provides services that can be used on multiple platforms such as Android, iOS, and the Web. Firebase is a recommendation because of its advantages for handling applications that prioritize functionality. In addition to the use of documentation that is also good, especially for the Android platform, integration of SDK (Standard Development Kit) is also effortless to do when using this service (Ilhami, 2017).

Firebase is a solution and alternative provided by Google to facilitate the work of mobile application developers. By using Firebase, it is hoped that mobile application developers will only focus on developing a reliable application without having to think about infrastructure, security, reliability, and server maintenance (Dermawan, 2017). So, by using Firebase, developers will focus on the reliability of applications that are developed and produce more reliable applications. The first product developed was Realtime Database, where developers can store and synchronize data to many users. Firebase provides several primary services that can be used for the development of this application, such as Firebase Authentication, Firebase Realtime Database, Firebase Cloud Firestore, and Firebase Storage (Siskarina, 2018).

2.1.7 Firebase Authentication

Most applications need to know the user's identity. By knowing the user's identity, the application can store user data securely in the cloud and provide the same personal experience on every device. Firebase Authentication provides back-end services with an easy-to-use SDK and libraries ready to be used to authenticate users to applications. Firebase Authentication aims to facilitate the construction of a secure authentication system while improving the login experience and orientation experience for users. Firebase Authentication supports quite many authentication methods using passwords, phone numbers, popular composite identity providers such as Google, Twitter, Facebook, Yahoo, Apple, GitHub, and Microsoft (Ilhami, 2017).

2.1.8 Firebase Realtime Database

Firebase Realtime Database is a database that is hosted in the cloud. Data that has been stored in JSON and synchronized in realtime to each connected user. Firebase Realtime Database is a NoSQL database and has different optimization and functionality compared to relational databases (Ilhami, 2017).

Firebase Realtime Database uses data synchronization every time the data changes. All connected devices will receive updates from the database within milliseconds. This can provide a collaborative and immersive experience. This database can be accessed directly through mobile devices or the web and not required an application server. Data security and validation can be arranged in the Firebase Realtime Database.

Data will also be stored on a local drive so that applications can still be used while offline. When the device is reconnected, the Realtime Database will re-synchronize changes that have been missed with the current server status. The Realtime Database is integrated with Firebase Authentication to provide an easy, secure, and intuitive authentication process for developers.

2.1.9 Firebase Storage

Firebase Storage is a reliable, simple, and cost-effective storage service. Firebase Storage is designed to help store and display photos and videos quickly and easily. This will make it easier to implement the system created because there is no need to use server-based programming. SDK for Storage adds Google security to file uploads and downloads. The SDK will automatically pause and resume transfers when connectivity is lost and reconnected, saving users time and bandwidth. Firebase Storage is also integrated with Firebase Authentication to provide easy, secure, and intuitive access control. Developers can set access controls for each file or collection of data (Ilhami, 2017).

2.1.10 Firebase Cloud Firestore

Cloud Firestore is a flexible database for mobile and web development. Cloud Firestore is similar to the Firebase Realtime Database, which allows us to store, synchronize, and query data quickly. Cloud Firestore will enable developers to create hierarchical and flexible data structures. This database uses data synchronization to update data on devices connected to the internet. By using Cloud Firestore, applications can write, read, detect, and perform data queries even if the device is offline and will synchronize again automatically when connected to the network. Data access in Firestore Cloud is also protected with Firebase Authentication.

2.1.11 Firebase Cloud Messaging

Firebase Cloud Messaging is a service from Firebase that is managed by Google. FCM is a platform that allows applications to send messages and notification notifications to users at no charge. Cloud Messaging provides a reliable and battery-saving connection between the server and the device. Notification messages can be immediately sent or schedule according to your wishes.

2.1.12 Kotlin

Kotlin is a programming language that runs on the Java Virtual Machine (JVM). Kotlin has a static language type that is the type of variable declared is static where after the variable is set, it cannot be changed again. Kotlin gives more focus on objects compared to Java because Kotlin offers functional programming with object-oriented programming. This makes Kotlin more attractive for developers to use. Kotlin wants to contribute to less code and has more productivity and alternative programming languages that are safer than Java.

Kotlin aims to be an improvement in the Java language. Kotlin is being promoted by Google Android and announced to be the official language for Android development at the 2017 Google I / O event. Kotlin is the third language that is supported by Android in addition to Java and C++. Kotlin is a programming language that is very suitable for developing Android applications. This language introduces many improvements for programmers, such as null-pointer safety, extension functions, and infix notation (Narang & Tuli, 2017).

2.2 Research Methodology

2.2.1 Literature Study Methods

In the literature study stage, data will be collected from a variety of supporting sources such as journals, theses, scientific works, reference books, and the internet related to the development of church location search applications and support and reinforce the theory of the proposed research. This method can be used to find out the theories needed in conducting research.

2.2.2 Method of Collecting Data

When collecting data, observations will be made by observing and recording related to the processes associated with the system designed. Observation is one of the data collection techniques by directly observing the object to be studied. Data collection procedures are carried out by taking data sources from churches in Yogyakarta.

Researchers will take data from all churches in the Yogyakarta area by visiting the church location directly. Data that can be taken from direct surveys to all churches in the form of church names, church contacts, church addresses, worship schedules, church photos, and church facilities. Researchers also got additional data from the internet in the form of church photos, church coordinates, church social media, and other additional info.

2.2.3 Software Requirements Analysis

The analysis process is a stage where analysis of system requirements is performed. This stage will be used to dig up as much information from users to suit their desires. At this stage, a user requirement will be generated. The results of the analysis are in the form of a software model written on the software requirements specification document.

2.2.4 Software Design

The design phase is the stage of casting thoughts and system design for the solution of existing problems. This stage will produce a use case diagram, ERD (Entity Relationship Diagram), interface design, as well as the structure and discussion of the data written in the software design description document.

2.2.5 Software Implementation

The implementation phase is the stage of writing program code. At this stage, the programmer translates software design into a programming language that matches the results of the analysis and system design. The programming language used in Kotlin with the Google Maps API. The result of the implementation phase is the source code that is ready to be executed and used.

2.2.6 Software Testing

The testing phase is the stage where testing of functionality has been made. The application system will be assessed, whether it is functionally following the specifications that have been made.

2.3 Literature Review

This research is research on church search applications that utilize Location Based Service and Firebase services. A comparison of research related to the search for a church like this is still very rare. Therefore, it can be one of the reference comparisons of other systems. Here are some things that can be used as a reference for this research.

Similar research conducted by Rahmani, Sushermanto, and Fitriyadi (2016) resulted in a service that could display natural attractions in the Banjar District. This study uses a Location-Based Service that is integrated directly with Google Maps to find a natural tourist attraction that is applied to Android smartphones. This study took analytical data at the Banjar District Culture Office for Youth Tourism and Sports and data obtained based on direct observations in the field.

Other research conducted by Ependi and Suyanto (2016) on the implementation of Location-Based Service for searching bus stops in Palembang resulted in an application that could show Transmusi bus stop points in Palembang according to user needs. Users can easily find and go to the nearest bus stop from its position because this application will display a road map to get to the location of the nearest bus stop. This application runs on mobile devices because users will find it easier and faster to access this application. Besides that, with a mobile device, users can access this application anytime and anywhere within the city of Palembang.

In research on the search for tourist sites in Bandung that utilize the Location Based Service method and the Google Maps API produces an application that can be used to provide information to users about info on tourist locations in Bandung. This application can provide information about tourist locations to users along with maps and route routes to these tourist sites. This application utilizes an integrated Global Positioning System (GPS), while maps and travel routes are implemented with the help of the Google Maps API (Putra et al., 2012).

Subsequent research discusses the application to search for camera service locations in Yogyakarta. This application outlines how to design Android mobile applications that utilize Firebase technology to search for camera service locations based on ratings. The results of this study have the facility to display several camera service search locations in the Yogyakarta area. The functionality of this research is displaying location profiles such as location via Google Maps, contacting directly via telephone, and email as well as giving a rating of these locations (Wahyujati, 2017).

Other similar research to find tourist sites in Semarang. The application is made with the available geographical information system-based features. The app can display a list of tourist attractions based on the type of tourist attractions and locations of attractions in Semarang. Tourist information displayed is divided into several categories, namely natural tourism, historical tourism, culinary tourism, and shopping tourism. Users can choose the type of tourist attractions as desired. The results of the study show that this application can display a list of tourist attractions, display tourist attraction details, and filter tourist attractions (Rahman et al., 2016).

In this study, the author will create an Android application for a church search in Yogyakarta. This application will utilize Location-Based Service technology, Global Positioning System, Google Maps API, and Firebase. The purpose of making this application is to provide information and help Catholics in Yogyakarta to find churches in Yogyakarta. The app that was built later will provide features of the location points of the church, provide church information, and be able to trace the route that can be taken. The advantage of this application is that it is easy to use, user friendly, and provides complete data in the form of church names, church addresses, church contacts, worship schedules, church social media, church facilities, church photos, church maps, and the latest information.

Table 2.1 Comparative Table

Comparative Element	**Rahmani et al. (2016)**	**Ependi dan Suyanto (2016)**	**Putra et al. (2012)**	**Rahman et al. (2016)**	**Wahyujati (2017)**	**Alvin Adhitama (2020)**
Application	Potential of Natural Tourism in the Regency of Banjar	Palembang BRT Transmission Stop Search	Bandung City Tourism	Search for tourist sites in Semarang	Camera service location search	Search for church locations in Yogyakarta
Platform	Android	Android	Android	Android	Android	Android
Database	MySQL	MySQL	MySQL	Firebase	Firebase	Firebase
Programming Language	Java	Java	Java	-	-	Kotlin
GPS	Trace the potential of nature tourism in the Banjar District	Track cultural information in Indonesia	Track tourist locations in the city of Bandung	Track tourist locations in the city of Semarang	Tracking the location of camera services in Yogyakarta	Track the location of the church in Yogyakarta
Integrated Development Environment	-	Eclipse	Eclipse	-	-	Android Studio
User Target	Local travelers	Local travelers	Traveler	Traveler	Traveler	Catholics, out-of-town tourists

CHAPTER 3. ANALYSIS AND DESIGN

3.1 System Analysis

Church Finder is an Android-based software that was built to find the location of a church in Yogyakarta. This application utilizes the reliability of Firebase to store and manage church data with Google Maps support for maps. The presence of this application is expected to increase the effectiveness of locating and obtaining church information in Yogyakarta.

3.2 Product Perspective

The application to search for a church is a system that can make it easy for Catholics both from Yogyakarta and outside the city to find churches around them. With this application, Catholics will get information on the location of the nearest church. The application that was built later will provide features of the location points of the church, provide church information, and display travel routes to the church. The Church Finder application offers information about the user's mobile position, then by knowing the location of the user can store and search for the location of the church around them.

This mobile software was developed to be used on the user's side of the Android operating system environment. This software was developed using the Kotlin programming language using the Android Studio programming environment. The database used is Firebase, and data management will be done through the Firebase website. The mobile application uses GPS on the device to get the coordinates of the location of the church. Users can easily find the location of the church by using the GPS feature found on the smartphone. This application is also assisted by using Google Maps provided by Google for the map of the direction to the church that is intended to make it easier for users to find out the position of the church. Another feature of Google that is used is Firebase Cloud Messaging, which can provide notifications in realtime to users.

Users will interact with the system via the GUI interface on an Android phone. This system is a client-server that is the user accesses the data contained on the webserver provided by Firebase, and the input data is stored in a database.

3.3 Problem Scope Analysis

The Church Finder application or church search mobile application was developed with the aim to:

1. Make it easy for Catholics to find and get info about the church in Yogyakarta.
2. Manage church data.
3. Display the location points of the church.
4. Display church information.

3.4 Data Needs Analysis

Some of the data needed in making a mobile application for church search in Yogyakarta are:

1. Church Data

   Church data required is in the form of church names, church addresses, worship schedules, church contacts, church social media, church facilities, church maps, church coordinates, and church photos.
2. User Data

   The required user data is the name, username, and email address that will be used as authentication to enter the application. Users can also save their favorite church so they can quickly get info on their favorite church.

3.5 External Interface Requirements

External interface requirements in the Church Finder application include user interface requirements, hardware interfaces, and software interfaces.

1. User interface

   Users interact with the interface that is displayed in the form of native apps on mobile devices. The interface is shown in the form of pages, list views, and maps on the mobile screen.
2. Hardware interface

   The hardware interface used to run the Church Finder application is:

    1. Smartphone with Android operating system.
    2. Smartphone with GPS.
    3. The smartphone can be connected to the internet.

3. Software interface

   The software needed to operate Church Finder application is as follow:

    1. Name : Android
       Source : Google
       Description : As an operating system for mobile devices.
    2. Name : Android Studio
       Source : Google
       Description : As a Church Finder application developer tool.
    3. Name : Firebase
       Source : Google
       Description : As a church data storage.
    4. Name : Adobe XD
       Source : Adobe
       Description : as a tool for designing User Interface and User Experience.
    5. Name : Google Maps API
       Source : Google
       Description : As an API for map creation within the application.

3.6 Product Functions

1. Login function

   This function is used by users to be able to enter and use applications.
2. Google login function

   This function is used by users to be able to log in and use
3. User register function

   This function is used by users to register to use the application.
4. Managing church data

It is a function used to manage church data. This management function is divided into church data added features, church data change features, and church data delete features.

4.1 Add data functions to the church

It is a function that is used to add data to the church that is done by the admin. Added data can be in the form of church names, church addresses, church contacts, worship schedules, church social media, church facilities, church photos, church maps.

4.2 The function of changing church data

It is a function used to change church data by the admin.

4.3 Delete church data function

It is a function used to delete church data.

5. The function of admin managing user data

   It is a function that is used for admin to manage user data. This management function is divided into delete user account functions and deactivate user account functions.

   5.1 Delete user account function

   This function is used to delete user accounts by the admin.

   5.2 The function of disabling user accounts

   This function is used to deactivate user accounts by the admin. But the account can be reactivated.

6. The function of managing user data

   It is a function that is used to manage user data. This management function is divided into password account reset function, email address change function, profile data change function, and password change function.

   6.1 Password account reset function

   Users use this function if they do not remember the password used. The user will enter the email address used to register the account then the system will send an email containing a verification link. Users only need to open the email then open the verification link. After that, the user can reset their account password.

   6.2 Email address change function

This function is used by users to change their email addresses. The application only asks the user to enter their new email address. After that, the application will change the user's email address.

6.3 Profile data change function

This function is used by users to change their profile data. Data that can be replaced are photo, name, and username.

6.4 Password change function

This function is used by users to change their passwords. The application will ask the user to enter a new password. Then the system will replace the user's old password with a new password.

7. The function displaying in church detail

   It is a function used to display details of the church. Data are shown in the form of church names, church addresses, church contacts, worship schedules, church social media, church facilities, church photos, church maps.

8. The added function of a favorite church

   It is a function used by users to add their favorite churches.

9. The function of searching for a church

   This function is used to search church data.

10. The function displays a map of the location of the church

    This function is used to display church locations in the form of maps.

11. The route display function

    It is a function used to help users display the route from the user's location to the church.

12. The function of displaying popular churches

    It is a function used to display popular churches in Yogyakarta.

13. The function of displaying a favorite church

    It is a function used to display the church that is a favorite of each user.

14. The function displays the trip route to the church

    This function is used to display the route to the church the user wants to go to.

3.7 Functional Software Requirements

3.7.1 User Use Case Diagram

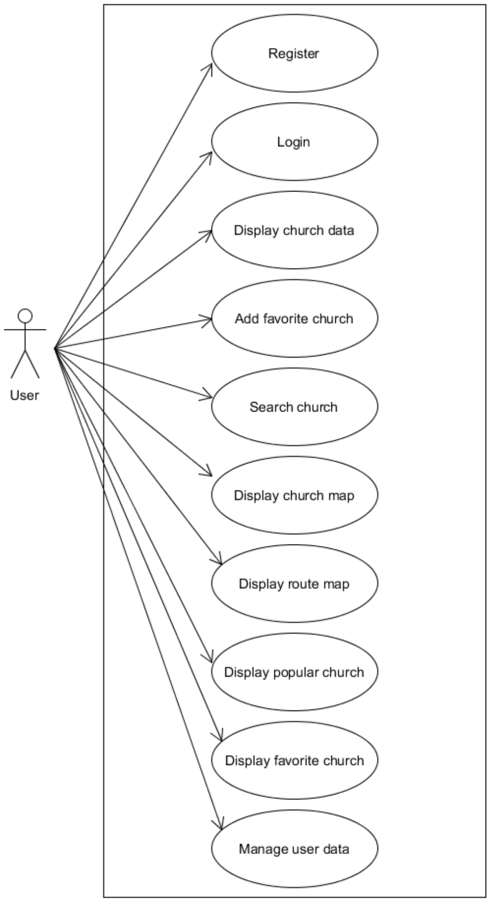


Figure 3.1 Church Finder User Use Case Diagram

Figure 3.1 illustrates the use case diagram of the Church Finder application. The actors in the use case diagram are users who use the Church Finder application. The first use case is done when the user will register to be able to enter the application. Use case register will ask the user to enter their name, username, photo, email address, and password to register in the Church Finder application. When the user does not enter all the data that must be filled in, the application will provide a warning to complete the existing data.

Table 3.1 Register Use Case Table

Use Case Name	:	Register User
Description	:	This use case explains how users register into the system.

Actor	:	User
Pre Condition	:	Users have never registered in the system.
Post Condition	:	User data is stored in the database. Enter the main page.
Basic Flow	:	1. The use case starts when there is a new user who wants to use the application. 2. The system displays the interface for registering. 3. The user enters the requested data. 4. The system checks the data entered by the user. E1: Invalid Data 5. The system saves new user data that has been entered. 6. The use case is complete.
Alternative Flow	:	
Error Flow	:	E1: Invalid Data E1 starting at step 4 basic flow 5. The system displays a warning that the data entered is invalid 6. Use case back to step 3

If the user already has an account in the application, the user will enter the use case login. Use case login will ask the user to enter a username and password. Users must have a username and password that has been registered in the Church Finder application. If you do not enter all the data that must be filled in or enter incorrect data, the app will provide a warning to complete the existing data.

Table 3.2 Login Use Case Table

Use Case Name	:	User Login
Description	:	This use case explains how users login to the system.
Actor	:	User
Pre Condition	:	Users are already registered in the system.
Post Condition	:	Enter the main page.
Basic Flow	:	1. The use case starts when the user wants to enter the application. 2. The system displays the interface for login. 3. The user enters the requested data. 4. The system checks the data entered by the user. E1: Invalid Data 5. The system gives access to users 6. The use case is complete.
Alternative Flow	:	
Error Flow	:	E1: Invalid Data

		E1 starting at step 4 basic flow 5. The system displays a warning that the data entered is invalid 6. Use case back to step 3

Users can also log in using a Google account. Users simply enter their Google account username and password. If an error occurs when entering data, then there will be a warning to enter the correct data. If the data entered is accurate, then the user can enter the application.

Table 3.3 Google Login Use Case Table

Use Case Name	:	Google Login
Description	:	This use case explains how users login in to the system using Google account.
Actor	:	User
Pre Condition	:	The user has a Google account.
Post Condition	:	Enter the main page.
Basic Flow	:	1. The use case starts when the user wants to enter the application. 2. The system displays the interface for Google login. 3. The user enters the requested data. 4. The system checks the data entered by the user. E1: Invalid Data 5. The system gives access to users 6. The use case is complete.
Alternative Flow	:	
Error Flow	:	E1: Invalid Data E1 starting at step 4 basic flow 5. The system displays a warning that the data entered is invalid 6. Use case back to step 3

Users can enter the application when the use case login or register is successful. In the app, there are eight use cases available. Use case show church data is a function to display the details of the data owned by the church. The data are shown in the form of the name of the church, church address, worship schedule, church facilities, church contacts, church social media, church maps, and photos of the church.

Table 3.4 Display Church Data Use Case Table

Use Case Name	:	Display Church Data
Description	:	This use case explains how the user sees detailed church information.
Actor	:	User

Pre Condition	:	The user has entered the application.
Post Condition	:	Users can view detailed church information
Basic Flow	:	1. The use case starts when the user wants to see detailed church information. 2. The user chooses the church they want to see. 3. The use case is complete.
Alternative Flow	:	
Error Flow	:	

Add a favorite church use case to run when choosing a church to add to be their favorite church. This function can be performed when the user is viewing church details from either a popular church of choice or another church. After the user chooses the church to be their favorite church, the church will be saved as the user's favorite church, and the user's chosen church can be viewed on the favorite church page.

Table 3.5 Add Favorite Church Use Case Table

Use Case Name	:	Add Favorite Church
Description	:	This use case explains how users add their favorite church.
Actor	:	User
Pre Condition	:	1. The user has entered the application. 2. Users already choose the church they want to see.
Post Condition	:	The user's favorite church is stored in the database.
Basic Flow	:	1. The use case starts when the user wants to add a favorite church 2. The user chooses the church they want to see. 3. Users press the add favorite button. 4. The use case is complete.
Alternative Flow	:	
Error Flow	:	

The use case shows the favorite church is used by users to see their favorite church. The favorite church menu is on the home page. When selecting the favorite church menu, the user will be taken to the favorite church page. The application will display a list of churches that are favorite users. Users can also view the details of their favorite church. Users can also delete their favorite church.

Table 3.6 Show User’s Favorite Church Use Case Table

Use Case Name	:	Show User’s Favorite Church

Description	:	This use case explains how users see their favorite church.
Actor	:	User
Pre Condition	:	1. The user has entered the application. 2. User already adds their favorite church.
Post Condition	:	The system shows the user's favorite church.
Basic Flow	:	1. The use case starts when the user wants to see their favorite church. 2. User open favorite church menu. 3. The system display user's favorite church. 4. The use case is complete.
Alternative Flow	:	
Error Flow	:	

Use cases display popular church is when the application shows several popular churches in Yogyakarta. Users can choose popular churches and see details of the church. The system will provide selected churches based on popularity in the Yogyakarta area.

Table 3.7 Display Popular Church Use Case Table

Use Case Name	:	Display Popular Church
Description	:	This use case explains how the user sees the popular church.
Actor	:	User
Pre Condition	:	The user has entered the application.
Post Condition	:	Users can view the popular church.
Basic Flow	:	1. The use case starts when the user wants to see the popular church. 2. The system shows the popular church. 3. The use case is complete.
Alternative Flow	:	
Error Flow	:	

Use case search church starts when the user selects the search menu on the main page. The application will open a search page and display a list of churches. Users can search for the church they want to search by the name of the church. The application will sort the church based on user input. If you have got the church that they are looking for, then the user can see the church details.

Table 3.8 Search Church Use Case Table

Use Case Name	:	Search Church
Description	:	This use case explains how users find the church they want.
Actor	:	User

Pre Condition	:	The user has entered the application.
Post Condition	:	Users find the church they want.
Basic Flow	:	1. The use case starts when the user wants to search for a church. 2. The user opens the search menu 3. The user enters the requested data. 4. The system searches the church entered by the user. E1: Invalid Data 5. The system shows the search result. 6. The use case is complete.
Alternative Flow	:	
Error Flow	:	E1: Invalid Data E1 starting at step 4 basic flow 5. The system displays a warning that the data entered is invalid 6. Use case back to step 3

Use case displays a map of the church running when the user opens the church map page. The application will display points of the location of the church in the form of a map. Users can see and choose which church they are looking for or go to. Users can open church details or choose to open a road route to the church they choose.

Table 3.9 Display Church Maps Use Case Table

Use Case Name	:	Display Church Map
Description	:	This use case explains how the users see the church location.
Actor	:	User
Pre Condition	:	The user has entered the application.
Post Condition	:	Users can view the church location on the maps.
Basic Flow	:	1. The use case starts when the user wants to see the church location. 2. Users open the maps menu. 3. The system shows the church location on maps. 4. The use case is complete.
Alternative Flow	:	
Error Flow	:	

The use case displaying a road route is a use case that runs when users want to show the road route to the church they want to go to. This function can be done through the church detail page or the church map page. Users simply press the button then the application will direct to the Google Maps application to help show the route of the road.

Table 3.10 Display Route Use Case Table

Use Case Name	:	Display Route
Description	:	This use case explains how the users see the route to the church.
Actor	:	User
Pre Condition	:	The user has entered the application.
Post Condition	:	Users can view the route to the church.
Basic Flow	:	1. The use case starts when the user wants to see the route to the church. 2. Users open the church detail page. 3. Users press the navigation button. 4. The system will show the route. 5. The use case is complete.
Alternative Flow	:	
Error Flow	:	

Use case managing user data is used by users to manage their data. Users can change their names, username, email, password, and photos. Users can also reset their password. If the user chooses to reset their password, then the application will send an email, and then the user will open their email to reset their password.

Table 3.11 Manage User Data Use Case Table

Use Case Name	:	Manage User Data
Description	:	This use case explains how users manage their data.
Actor	:	User
Pre Condition	:	Users are already registered in the system.
Post Condition	:	Users can edit their data.
Basic Flow	:	1. The use case starts when the user wants to edit their data. 2. The user opens the profile menu. 3. The user chooses the data that they want to edit. A1: User choose edit profile A2: User choose edit email A3: User choose edit password 4. The system saves the data. E1: Invalid Data 5. The use case is complete.
Alternative Flow	:	A1: Edit Profile A1 starting at step 3 basic flow 1. The system opens the interface to edit the user profile. 2. User input the requested data.

		3. Use case back to step 4 basic flow. A1: Edit Email A1 starting at step 3 basic flow 1. The system opens the interface to edit the user profile. 2. User input the requested data. 3. Use case back to step 4 basic flow. A1: Edit Password A1 starting at step 3 basic flow 1. The system opens the interface to edit the user profile. 2. User input the requested data. 3. Use case back to step 4 basic flow.
Error Flow	:	E1: Invalid Data E1 starting at step 4 basic flow 5. The system displays a warning that the data entered is invalid 6. Use case back to step 3 basic flow

3.7.2 Admin Use Case Diagram

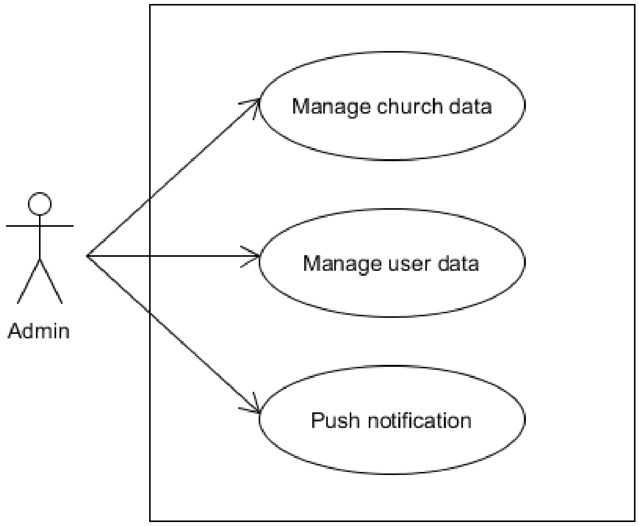


Figure 3.2 Church Finder Admin Use Case Diagram

Figure 3.2 is a function that can be done by the admin. The actor in this use case is admin. The first use case is managing church data. Use managing church data can be done when the admin opens the Firebase database. Admin can add data, change data, and delete church data. The data in the database are in the form of the name of the church, church address, church contact, worship schedule, church social media, church facilities, church photos, and church maps.

Table 3.12 Manage Church Data Use Case Table

Use Case Name	:	Manage Church Data
Description	:	This use case explains how admin manages church data.
Actor	:	Admin
Pre Condition	:	Admin opens the Firebase console.
Post Condition	:	Admin manages church data.
Basic Flow	:	1. The use case starts when the admin wants to manage church data. 2. Admin open Firebase console 3. Admin manages the church data 4. The system saves the data. 5. The use case is complete.
Alternative Flow	:	
Error Flow	:	

Use case managing user data is done when the admin wants to manage user accounts. When the admin opens the Firebase database, the admin can manage user data by deactivating the account and deleting the user account. By removing a user's account, the user cannot enter the application again if he does not register again. By disabling the account, the user cannot enter the app, but the admin can activate it back so that the user can re-enter the application.

Table 3.13 Manage User Data Use Case Table

Use Case Name	:	Manage User Data
Description	:	This use case explains how admin manages admin data.
Actor	:	Admin
Pre Condition	:	Admin opens the Firebase console.
Post Condition	:	Admin manages admin data.
Basic Flow	:	1. The use case starts when the admin wants to manage admin data. 2. Admin open Firebase console 3. Admin manages the church data 4. The system saves the data. 5. The use case is complete.
Alternative Flow	:	
Error Flow	:	

Use case push notification is done when the admin wants to send a notification message to the user. When the admin opens the Firebase console, the admin can easily send messages quickly and easily.

Table 3.14 Push Notification Use Case Table

Use Case Name	:	Push Notification
Description	:	This use case explains how admin push notification to the users.
Actor	:	Admin
Pre Condition	:	Admin opens the Firebase console.
Post Condition	:	Admin sends a notification to the users.
Basic Flow	:	1. The use case starts when the admin wants to send a notification to the users. 2. Admin open Firebase console 3. Admin makes the notification. 4. The system sends the notification. 5. The use case is complete.
Alternative Flow	:	
Error Flow	:	

3.8 Entity Relationship Diagram (ERD)

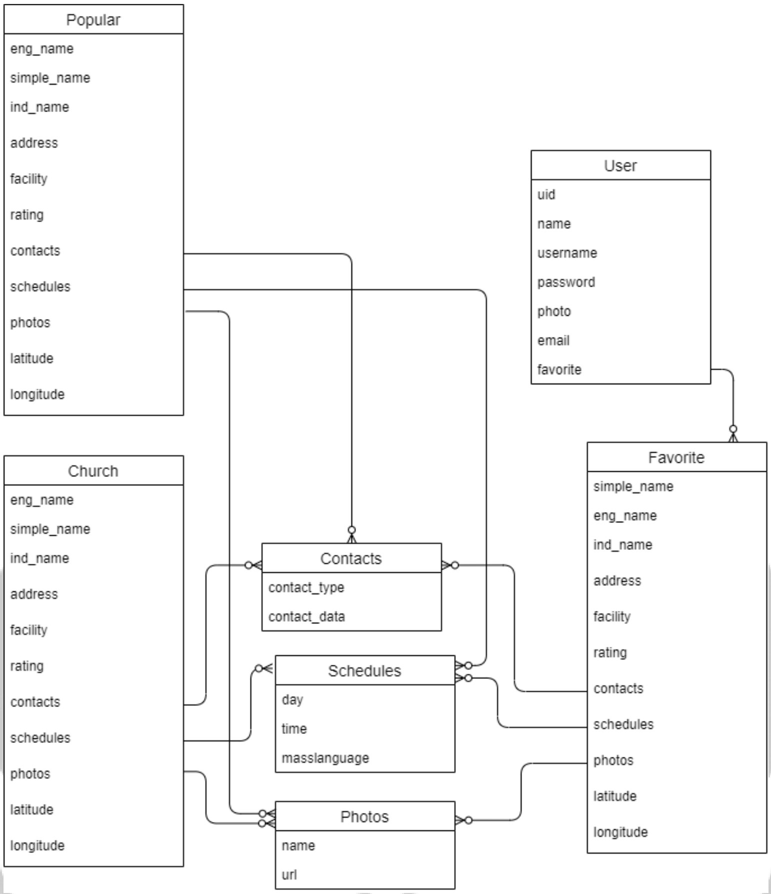


Figure 3.3 Entity Relationship Diagram Church Finder

ERD (Entity Relationship Diagram) is a model for explaining relationships between data in a database based on data base objects that have relationships between relations. ERD is used to model data structures and relationships between data, to illustrate it uses several notations and symbols. This diagram makes it easier to develop a system because in the ERD there is already a general description and details of a system designed.

3.9 Software Architecture Design

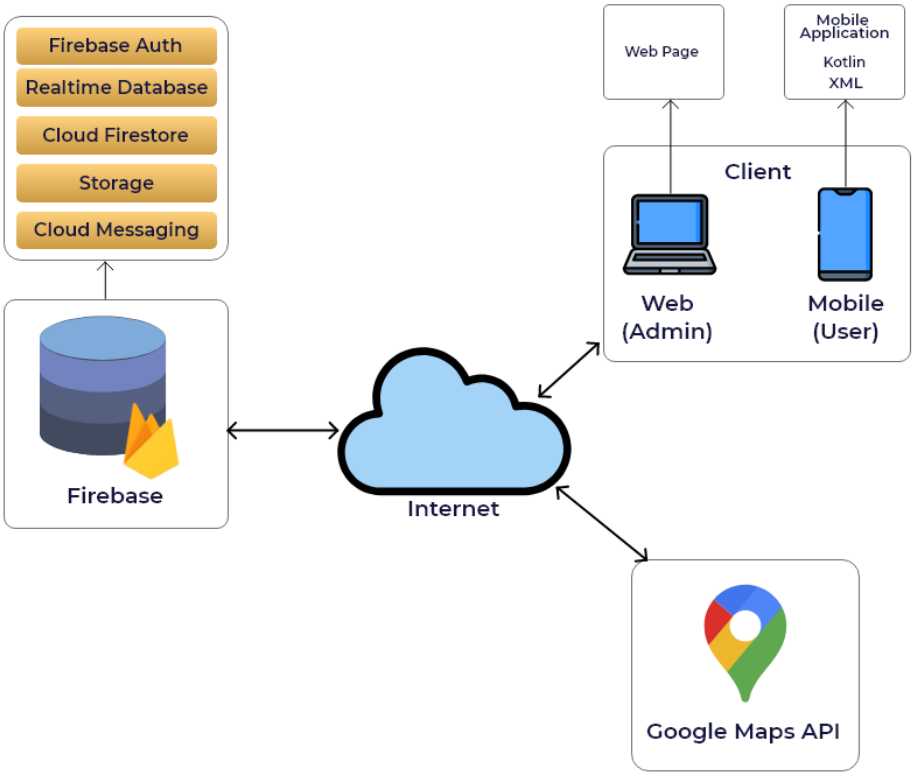


Figure 3.4 Software Architecture Design

Figure 3.4 is an architectural design of the software built. There are two actors, namely user and admin. The user only operates the mobile application, while the admin can manage the web. Each actor has their respective functions that can be performed on a mobile app or a web page.

Users can operate using the available mobile applications, and users can log into the app, manage user data, display church data, search for churches, and display travel routes. The mobile application uses GPS on the device to get the coordinates of the exact location of the church. Users can easily find the location of the church needed by using the GPS feature found on the smartphone. This mobile application provides a church search feature using the help of Google Maps.

Admin can enter the system by using a web browser to do some management, namely church data management, and user management. Admin can add, change, and delete church data.

All data is stored in a database, so to connect the mobile application with the database requires an API so that the data can only be accessed with special authentication.

3.10 Database Design

Table 3.15 Church Class

Attribute Name	Data Type	Description
eng_name	String	English name of the church
ind_name	String	Indonesian name of the church
simple_name	String	Short name of the church
address	String	Address of the church
contacts	String	Contacts of the church
schedules	String	Mass schedule of the church
photos	String	Photos of the church
display	String	Display picture of the church
poster	String	Poster image of the church
latitude	Double	Latitude coordinate of the church
longitude	Double	Longitude coordinate of the church

Table 3.2 Favorite Church Class

Attribute Name	Data Type	Description
uid	String	User uid
eng_name	String	English name of the church
ind_name	String	Indonesian name of the church
simple_name	String	Short name of the church
address	String	Address of the church
contacts	String	Contacts of the church
schedules	String	Mass schedule of the church

photos	String	Photos of the church
display	String	Display picture of the church
poster	String	Poster image of the church
latitude	Double	Latitude coordinate of the church
longitude	Double	Longitude coordinate of the church

Table 3.17 Popular Class

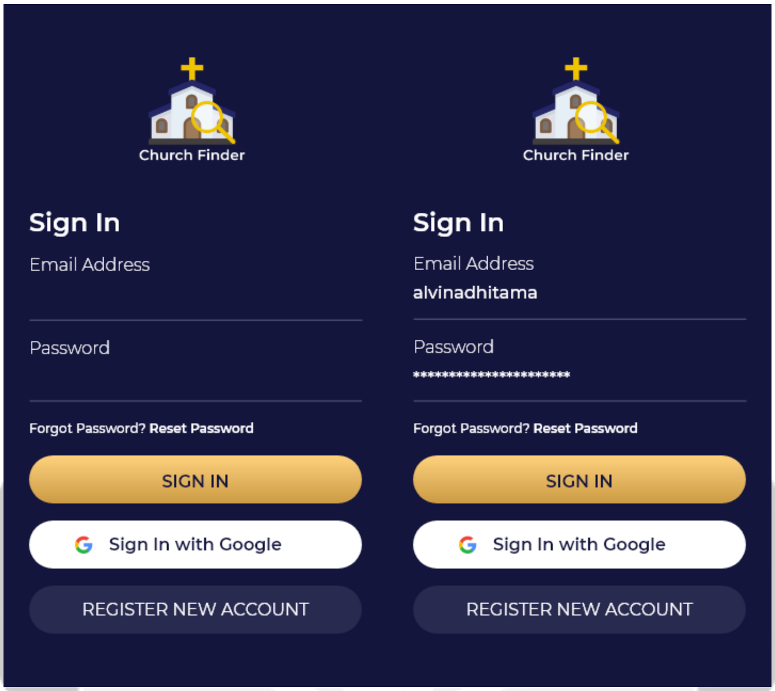
Attribute Name	Data Type	Description
eng_name	String	English name of the church
ind_name	String	Indonesian name of the church
simple_name	String	Short name of the church
address	String	Address of the church
contacts	String	Contacts of the church
schedules	String	Mass schedule of the church
photos	String	Photos of the church
display	String	Display picture of the church
poster	String	Poster image of the church
latitude	Double	Latitude coordinate of the church
longitude	Double	Longitude coordinate of the church

Table 3.18 User Class

Attribute Name	Data Type	Description
uid	String	Unique key of the user
name	String	Name of the user
email	String	Email of the user
username	String	Username of the user
photo	String	User profile photo
account	String	User account type

3.11 Interface Design

3.11.1 Login Page Interface Design



Email Address
alvinadhitama
Password
****

Figure 3.5 Login Page Interface Design

The interface design in Figure 3.5 is used for logging into the Church Finder application system. To get access to enter the system, the user must enter an email address and password that has been registered in the textbox provided. When the sign-in button is pressed, the system will check the email address and password that have been entered with the data contained in the database. If the data entered is correct, then the user can enter the system. Conversely, if the email and password are empty or wrong, then the application will give a warning.

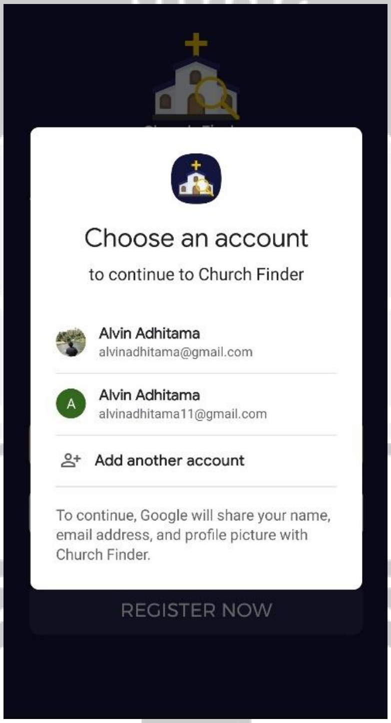
In the interface design in Figure 4.5, there is also a button to sign in through a Google account. Users simply press the sign-in button through Google then the application will direct the user to fill in their Google account email and password data.

Table 3.19 Login Functional Requirements Table

Description	:	The login function must enable the user to enter the system using email and password.

Validity Check	:	▪ Email: alphanumeric, is a combination of numbers, letters, and uses the email format. ▪ Password: alphanumeric, which has a minimum of 6 characters. Allowed characters are letters, digits, and special characters. Space not allowed.
Rational	:	This function provides security and limits user accounts so that only authenticated users can enter the system.

3.11.2 Login with Google Page Interface Design



Choose an account

to continue to Church Finder

Alvin Adhitama
alvinadhitama@gmail.com

Alvin Adhitama
alvinadhitama11@gmail.com

Add another account

To continue, Google will share your name, email address, and profile picture with Church Finder.

Figure 3.6 Reset Password Page Interface Design

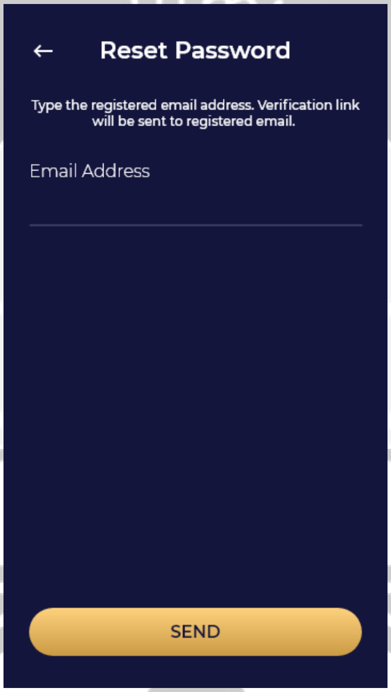
The Google login screen page in Figure 3.6, is a login display if the user chooses to use their Google account to enter the system. Firebase Authentication also provides login facilities through Google. Users simply select the Google account they want to use.

Table 3.20 Google Login Functional Requirements Table

Description	:	The login function must enable the user to enter the system using a Google account.

Validity Check	:	▪ Email: alphanumeric, is a combination of numbers, letters, and uses the email format. ▪ Password: alphanumeric, which has a minimum of 6 characters. Allowed characters are letters, digits, and special characters. Space not allowed.
Rational	:	This function provides security and limits user accounts so that only authenticated users can enter the system.

3.11.3 Reset Password Page Interface Design



Type the registered email address. Verification link will be sent to registered email.

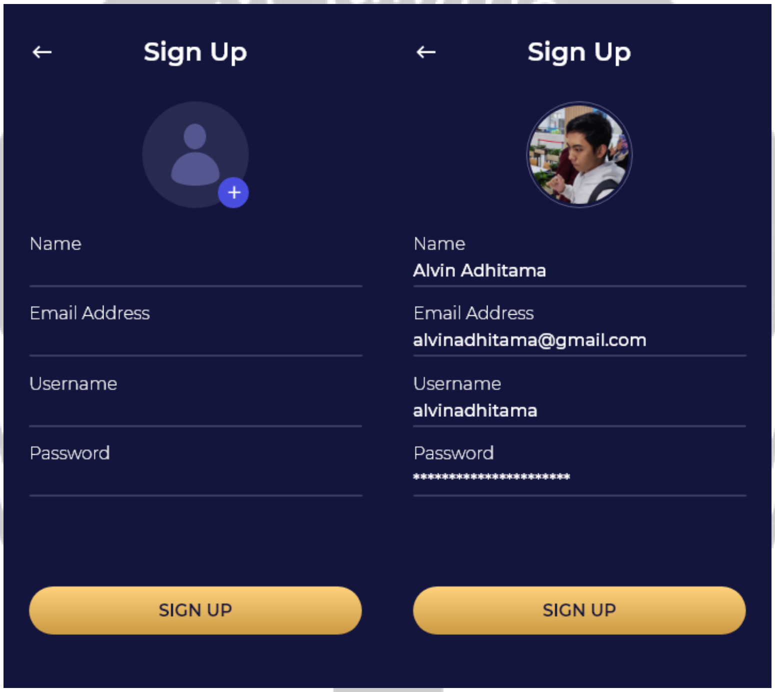
Figure 3.7 Reset Password Page Interface Design

The interface design in Figure 3.7 is the page used by users to reset their account passwords. The system will ask the user to enter an email address that is already registered. The send button is used to send a verification link to the email address entered by the user. If the email address provided is already registered, then the verification link will get to the email user. If the email address provided is not registered, then the application will give a warning and then ask the user to re-enter the email address.

Table 3.21 Reset Password Functional Requirements Table

Description	:	The login function must enable the user to their password.
Validity Check	:	▪ Email: alphanumeric, is a combination of numbers, letters, and uses the email format.
Rational	:	This function makes it easy for users to forget their password and want to reset their password.

3.11.4 Register Page Interface Design



Name
Alvin Adhitama

Email Address
alvinadhitama@gmail.com

Username
alvinadhitama

Password
****

Figure 3.8 Register Page Interface Design

The interface design in Figure 3.8 is used to add a new user. The data needed is a photo, name, username, email address, and password. To add pictures, users can specify photos to be selected via the gallery or take pictures directly through the camera. After entering the data, the user can press the sign-up button to save all data that has been entered. The system will check the user data that has been entered, and if the data is appropriate, then the system will add the user data into the database. If it is incorrect or blank data, the system will display a warning.

Table 3.22 Register Functional Requirements Table

Description	:	The register function must enable the user to register into the system. User attributes that are registered into the system contain the name, username, photo, email address, and password.
Validity Check	:	▪ Name: consists of letters with spaces between are allowed. ▪ Username: alphanumeric, is a combination of letters and numbers. Spaces and special characters are not allowed. ▪ Email: alphanumeric, is a combination of numbers, letters, and uses the email format. ▪ Password: alphanumeric, which has a minimum of 6 characters. Allowed characters are letters, digits, and special characters. Space not allowed. ▪ Photo: photo in jpg/png format.
Rational	:	This function provides security and limits user accounts so that only authenticated users can enter the system. This function provides the ability to add user data into the system.

3.11.5 Home Page Interface Design

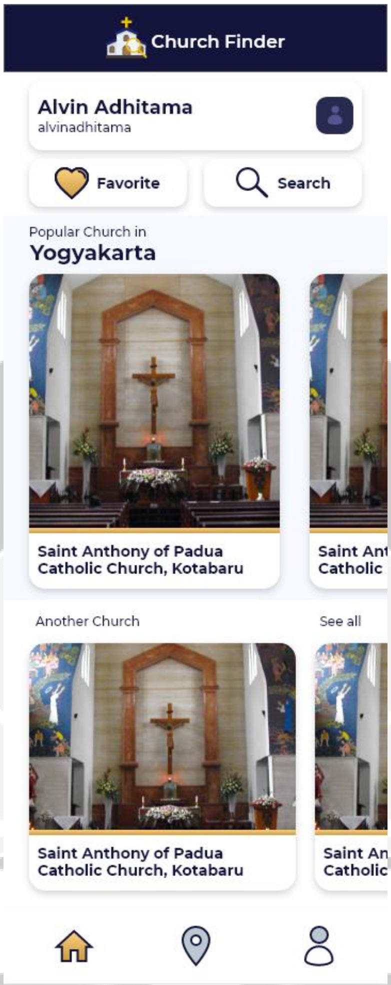


Figure 3.9 Home Page Interface Design

The interface design in Figure 3.9 is a page that opens after the user has successfully logged in or registered. When the main menu page appears, the user will be faced with several menus. On the main menu page, two list items are featuring popular churches in Yogyakarta and other churches. The favorite button is used by the user to view their favorite church. The search button is used to be used to find what they want. The map menu is used to show the location of churches in the Yogyakarta area. Menu for displaying user-profiles and application settings. Users can also view all church data through the see all button.

Table 3.23 Home Page Functional Requirements Table

Description	:	This function must enable the app to display the list of church data and popular church data in the system.

Validity Check	:	None
Rational	:	This function gives information to the user if they want to see a list of churches and a list of popular churches.

3.11.6 Maps Page Interface Design

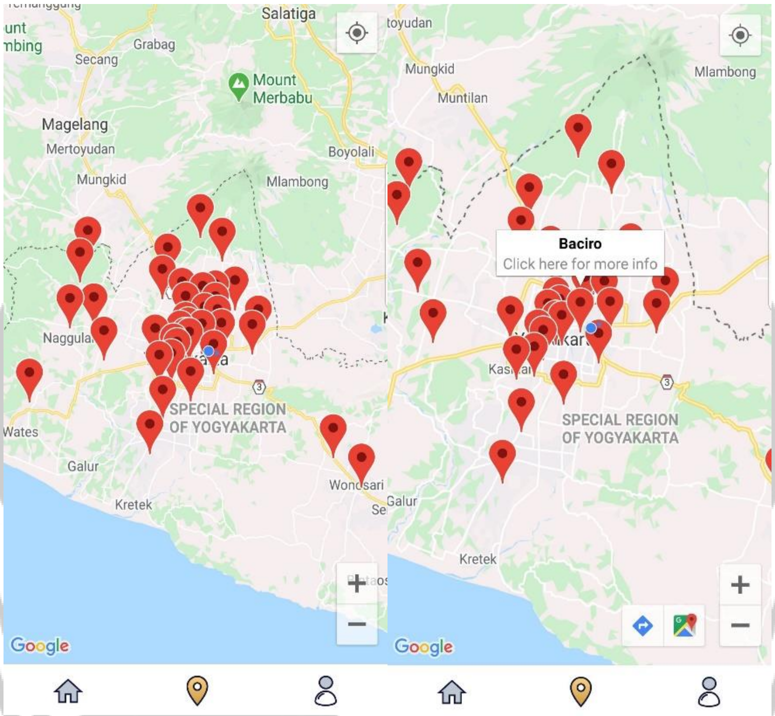


Figure 3.10 Maps Page Interface Design

The interface design in Figure 3.10 is a map page used to display church locations in Yogyakarta. Users can search for the church closest to their position. The data needed is the latitude and longitude of each church that is stored in the application and then displayed in the form of markers. Users can use maps on this page, like using the Google Maps application because this application uses the API from Google Maps to display maps and travel routes.

Users can search their current position, zoom in or zoom out, and pan the map. If the user wants to view information about the church they chose, then the user simply presses the location of the church, then the application will direct the user to the church details page. If the user wants to display the route to get to the church they want, the user simply presses the arrow button at the bottom that will point to the Google Maps application.

Table 3.24 Maps Page Functional Requirements Table

Description	:	Function show locations must enable the app to display the church location on the map.
Validity Check	:	None
Rational	:	This function gives information to the user if they want to see the church location on the map.

3.11.7 Profile Page Interface Design

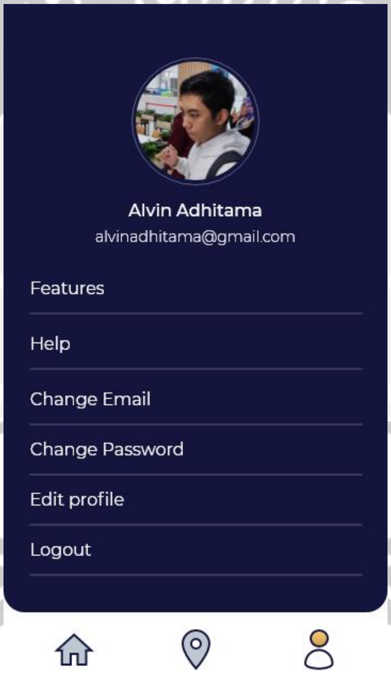


Figure 3.11 Profile Page Interface Design

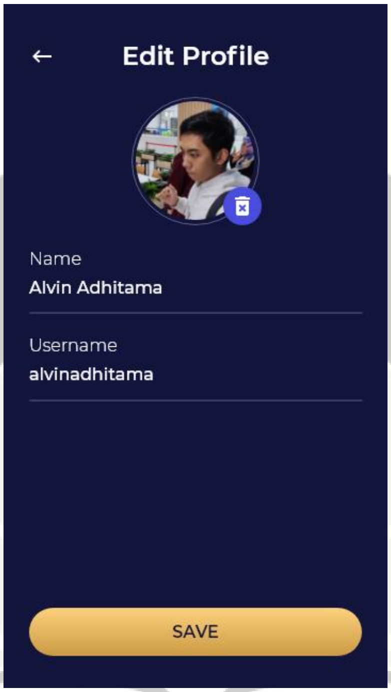
The interface design in Figure 3.11 is the user's profile page. In the profile page, several functions can be used. Users can find out the features contained in the Church Finder application, log out of the application, change email addresses, change passwords, and edit user profiles.

Table 3.25 Logout Functional Requirements Table

Description	:	The logout function must enable the user to log out when it has finished using the software.
Validity Check	:	None

Rational	:	This function provides security in terms of authentication because the system can find out who has used the application.

3.11.8 Edit Profile Interface Design



Edit Profile

Name
Alvin Adhitama

Username
alvinadhitama

Figure 3.12 Edit Profile Interface Design

The interface design in Figure 3.12 is used to change user data. Data that can be changed are photos, names, and usernames. To change photos, users can specify photos to be selected via the gallery or take pictures directly through the camera. After replacing data, the user can press the save button to save the data in the database. The system will check the data that has been entered, and if the data is appropriate, then the system will store data. If there is an error, the system will give a warning. The profile edit menu cannot be used to change the password, so users must enter the password reset menu to change their password.

Table 3.26 Edit Profile Functional Requirements Table

Description	:	The function of editing user data must enable the user to make changes to the data in the system.
Validity Check	:	▪ Name: consists of letters with spaces between are allowed.

		▪ Username: alphanumeric, is a combination of letters and numbers. Spaces and special characters are not allowed. ▪ Photo: photo in jpg/png format.
Rational	:	This function provides the ability to manage user data; in this case, changing user data if there is an error in inputting or data changes.

3.11.9 Edit Email Interface Design

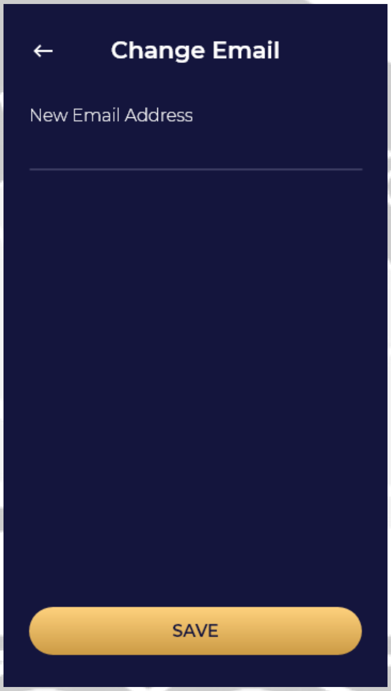


Figure 3.13 Edit Email Interface Design

The interface design in Figure 3.13 is used to change user e-mail. To change e-mail, the user can enter a new e-mail. The save button is used to save a user's new email into the database. If an error occurs, the system will provide a warning and ask for re-input.

Table 3.27 Edit Email Functional Requirements Table

Description	:	The user email editing function must enable the user to make changes to the user's email in the system.
Validity Check	:	▪ Email: alphanumeric, is a combination of numbers, letters, and uses the email format.

Rational	:	This function provides the ability to manage user data, in this case, changing user email if the user wants to change their email address.

3.11.10 Edit Password Interface Design

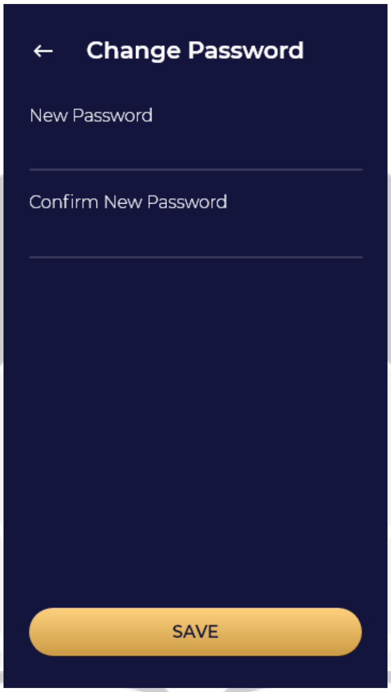


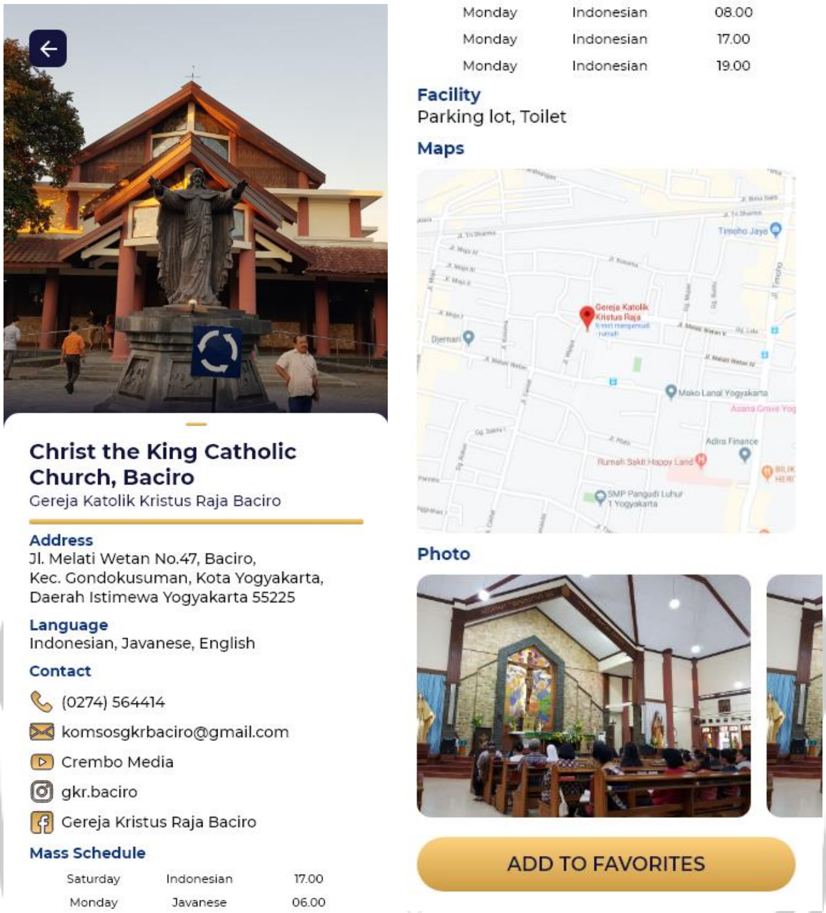
Figure 3.14 Edit Password Interface Design

The interface design in Figure 3.14 is used to change the user's password. The application will ask the user to enter a new password and then re-request the new password for verification. After that, the user can press the save button to save their new password. If an error occurs, the system will give a warning.

Table 3.28 Edit Password Functional Requirements Table

Description	:	The user password editing function must enable the user to make changes to the user's password in the system.
Validity Check	:	▪ Password: alphanumeric, which has a minimum of 6 characters. Allowed characters are letters, digits, and special characters. Space not allowed.
Rational	:	This function provides the ability to manage user data, in this case, changing user password if the user wants to change their password

3.11.11 Church Detail Interface Design



Christ the King Catholic Church, Baciro

Gereja Katolik Kristus Raja Baciro

Address

Jl. Melati Wetan No.47, Baciro,
Kec. Gondokusuman, Kota Yogyakarta,
Daerah Istimewa Yogyakarta 55225

Language

Indonesian, Javanese, English

Contact

☎ (0274) 564414

komsosgkrbaciro@gmail.com

Crembo Media

gkr.baciro

Gereja Kristus Raja Baciro

Mass Schedule

Saturday	Indonesian	17.00
Monday	Javanese	06.00
Monday	Indonesian	08.00
Monday	Indonesian	17.00
Monday	Indonesian	19.00

Facility

Parking lot, Toilet

Maps

Photo

Figure 3.15 Detail Church Interface Design

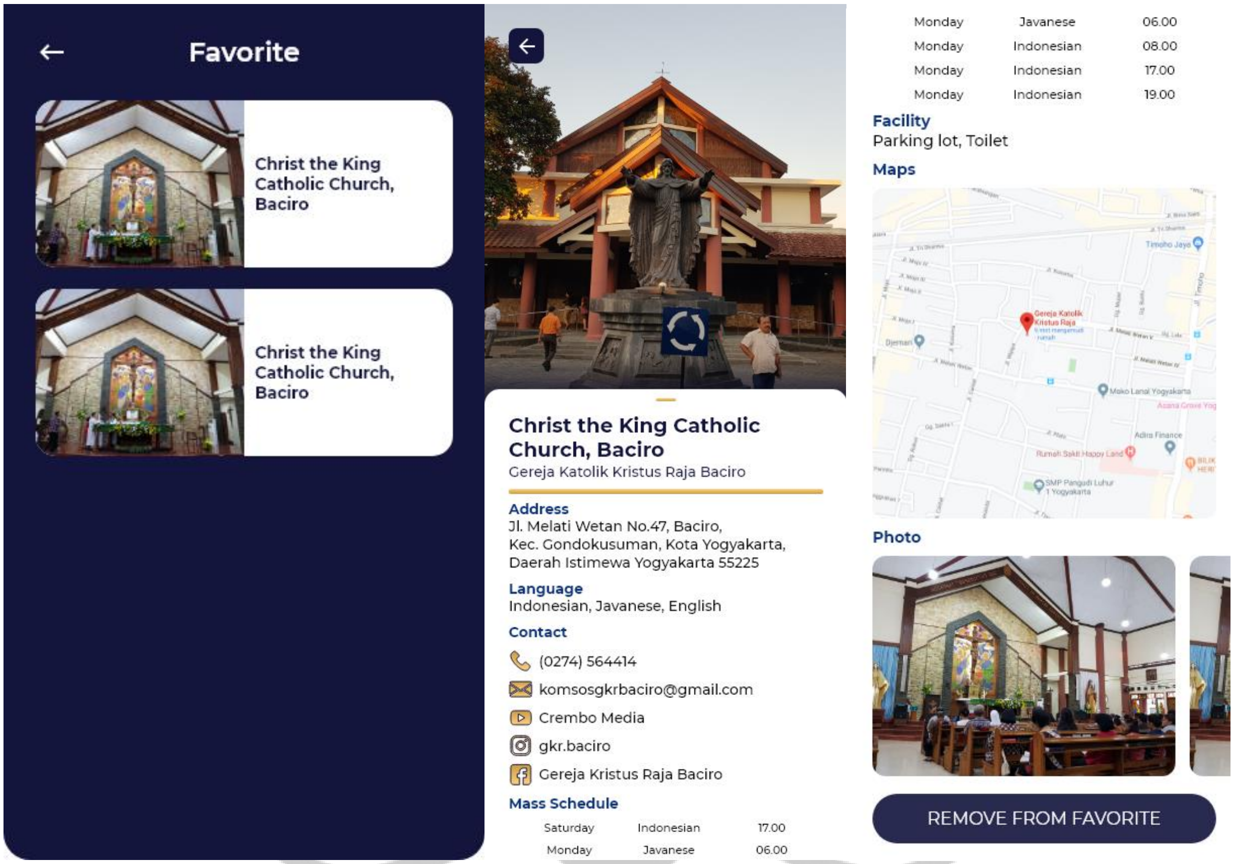
The interface design in Figure 3.15 shows more specific church information in the form of church names, church addresses, church contacts, church social media, worship schedules, church facilities, church maps, and church photos. Then there is a button that can be used to add the church to the user's favorites. After pressing the button, users can easily find their favorite church on the favorite page.

Table 3.29 Church Detail Functional Requirements Table

Description	:	The function of displaying church details must enable the application to display church data in the system.
Validity Check	:	None

Rational	:	This function provides information to the user if the user wants to see detailed church data information.

3.11.12 Favorite Page and Detail Favorite Interface Design


Favorite
Christ the King Catholic Church, Baciro
Christ the King Catholic Church, Baciro
Maps
Photo
REMOVE FROM FAVORITE

Christ the King Catholic Church, Baciro
Gereja Katolik Kristus Raja Baciro

Address
Jl. Melati Wetan No.47, Baciro,
Kec. Gondokusuman, Kota Yogyakarta,
Daerah Istimewa Yogyakarta 55225

Language
Indonesian, Javanese, English

Contact
(0274) 564414
komsosgkrbaciro@gmail.com
Crembo Media
gkr.baciro
Gereja Kristus Raja Baciro

Mass Schedule

Saturday	Indonesian	17.00
Monday	Javanese	06.00
Monday	Javanese	06.00
Monday	Indonesian	08.00
Monday	Indonesian	17.00
Monday	Indonesian	19.00

Facility
Parking lot, Toilet

Figure 3.16 Favorite Page and Detail Favorite Interface Design

The interface design 3.16 displays a list of users' favorite churches as well as detailed pages of favorite pages. Favorite pages will show churches that have been favored by users. Users can also display church details specifically. Data that can be displayed is the name of the church, church address, church contact, church social media, worship schedule, church facilities, church maps, and church photos. Then there is a button that can be used to delete the church from the user's favorite.

Table 3.30 Church Favorite Functional Requirements Table

Description	:	The function of displaying the user's favorite church must enable the application to display the user's favorite church data.
Validity Check	:	None
Rational	:	This function provides information to the user if the user wants to see their favorite church data information.

3.11.13 Display All Church Interface Design

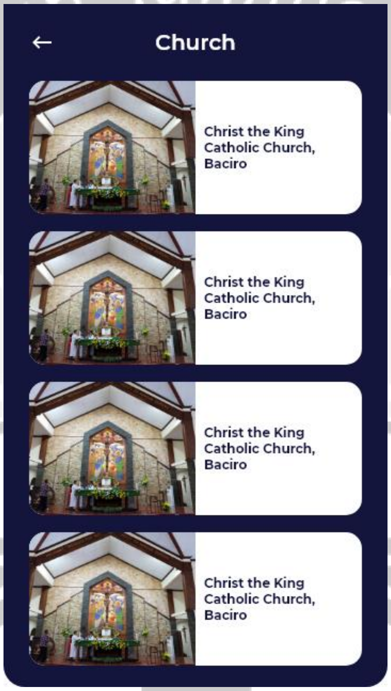

Church
Christ the King Catholic Church, Baciro
Christ the King Catholic Church, Baciro
Christ the King Catholic Church, Baciro
Christ the King Catholic Church, Baciro

Figure 3.17 Display All Church Interface Design

The interface design in Figure 3.17 is the page used to display all the church data contained in the system. The application will display a list of data from the church. Users can see and search for the church according to their wishes. Users can also view detailed information from the church by pressing the view on the church of their choice.

Table 3.31 All Church Functional Requirements Table

Description	:	The function of displaying all church must enable the application to display the list of church data.

Validity Check	:	None
Rational	:	This function provides information to the user if the user wants to see the list of all churches.

3.11.14 Search Church Interface Design

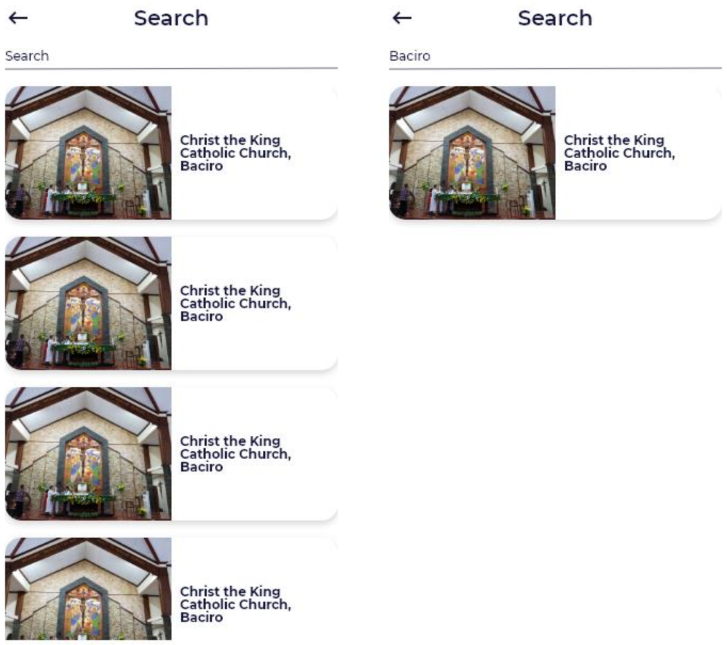

Search
Search
Christ the King Catholic Church, Baciro
Christ the King Catholic Church, Baciro
Christ the King Catholic Church, Baciro
Christ the King Catholic Church, Baciro
Search
Baciro
Christ the King Catholic Church, Baciro

Figure 3.18 Search Page Interface Design

The interface design 3.18 is used by users to find the church they want. Church search can be done by entering the name of the church in the search section, and the system will sort the church data based on user input. After getting the church data that the user wants, then the user can see the complete church information by pressing a view on the church search results.

Table 3.32 Search Church Functional Requirements Table

Description	:	The church search function must enable users to search church data in the system.
Validity Check	:	▪ Church name: consists of letters.

Rational	:	This function makes it easy for users to find church data when users want to find out church information.

LIST OF REFERENCES

Anwar, S., Nugroho, I., & Lestariningsih, E. (2013). Perancangan dan Implementasi Aplikasi Mobile Semarang Guidance pada Android. *Dinamik*.

Apriyani, M. E., Giovanny, R., & Haris, P. Y. (2012). Sistem Pelacakan Posisi Kapal Berbasis Mobile Android dan Web Server. *Jurnal Integrasi*, *4*.

Darma, G. N., WP, S. P., & Anindito, K. (2012). Perancangan Aplikasi Mobile City Directory Yogyakarta Berbasis Android. *Seminar Nasional Teknologi Informasi Dan Komunikasi*, 136.

Dermawan, I. G. N. (2017). Pembangunan Aplikasi Mobile Pencarian Lokasi Nonton Bareng Berbasis Location Based Service.

Ditjen Kependudukan dan Pencatatan Sipil. (2019). Jumlah Penduduk Menurut Agama Semester I 2019. Retrieved January 24, 2020, from https://kependudukan.jogjaprov.go.id/olah.php?module=statistik&periode=12&jenisdata=penduduk&berdasarkan=agama&prop=34&kab=&kec=00

Ilhami, M. (2017). Pengenalan Google Firebase Untuk Hybrid Mobile Apps Berbasis Cordova. *Jurnal IT CIDA*.

Juhara, P. Z. (2016). Panduan Lengkap Pemrograman Android. In *Panduan Lengkap Pemrograman Android*.

Khedkar, S., Thube, S., Estate, W. I., W, N. M., & Naka, C. (2017). Real Time Databases for Applications. *International Research Journal of Engineering and Technology(IRJET)*.

Mahdia, F., & Noviyanto, F. (2013). Pemanfaatan Google Maps API untuk Pembangunan Sistem Informasi Manajemen Bantuan Logistik Pasca Bencana Alam Berbasis Mobile Web. *Jurnal Sarjana Teknik Informatika*.

Munir, M. W., Omair, S. M., & Haque, M. Z. U. (2015). An Android based Application for Determine a Specialized Hospital Nearest to Patient Location. *International Journal of Computer Applications*. https://doi.org/10.5120/20776-3316

Narang, J., & Tuli, E. S. (2017). Review Study on New Era of Android Kotlin. *International Journal of Technology and Computing*, *3*(8).

Nazruddin Safaat H. (2013). Android : Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android (Edisi Revisi). In *Android*.

Paraya, G. R., & Tanone, R. (2018). Penerapan Firebase Realtime Database Pada Prototype Aplikasi Pemesanan Makanan Berbasis Android. *Jurnal Teknik Informatika Dan Sistem Informasi*. https://doi.org/10.28932/jutisi.v4i3.870

Pratami, R. D., Bettiza, M., & Kurniawan, H. (2014). Aplikasi Pencarian Tempat Makan menggunakan Location Based Service Pada Android. *Jurnal Informatika Universitas Maritim Raja Ali Haji*.

Putra, A., Tambunan, T., & Ramadhan, K. (2012). Aplikasi Wisata Kota Bandung Menggunakan Metode Location-Based Services (LBS) pada Android. *Jurnal Telkom University*.

Rahman, M., Christyono, Y., & Santoso, I. (2016). Rancang Bangun Aplikasi MyTrip untuk Pencarian Lokasi Wisata di Wilayah Semarang dalam Perangkat Android Berbasis GIS (Geographic Information System). *Universitas Diponegoro*, *5*.

Rahmani, Fitriyadi, & Sushermanto. (2016). Aplikasi Location Based Service Potensi Wisata Alam Kabupaten Banjar Berbasis Android. *Jurnal Ilmiah Komputer*, *12*.

Rahmi, A. (2017). *Sistem Informasi Geografis Pencarian Dokter Berbasis Android dengan Google Maps API dan Firebase Cloud Messaging*. Universitas Udayana.

Saputra, I. U., Sinsuw, A., & B.N. Najoan, X. (2017). Pengembangan Aplikasi Location Based Service Pariwisata berbasis Android Studi Kasus Kabupaten Toraja Utara. *Jurnal Teknik Informatika*. https://doi.org/10.35793/jti.12.1.2017.17854

Segara, R., & Subari, S. (2017). Sistem Pemantauan Lokasi Anak Menggunakan Metode Geofencing pada Platform Android. *Jurnal Teknologi Dan Manajemen Informatika*. https://doi.org/10.26905/jtmi.v3i1.629

Shalahudin, M., & Rosa, A. . (2016). Rekayasa Perangkat Lunak Terstruktur dan Berrientasi Objek. *Informatika Bandung*.

Siskarina. (2018). Pembangunan Aplikasi Mobile Sistem Pengingat Aktivitas Mahasiswa Universitas Atma Jaya Yogyakarta.

Statistik, B. P. (2019). Kewarganegaraan, SuKu Bangsa, Agama, dan Bahasa Sehari-hari Penduduk Indonesia Hasil Sensus Penduduk 2010. In *Badan Pusat Statistik Indonesia*. https://doi.org/10.1017/CBO9781107415324.004

Subastian, A. (2018). *Pengembangan Sistem Informasi Bursa Kerja Khusus (BKK) Berbasis Android di SMK YPKK 1 Sleman*. Universitas Negeri Yogyakarta.

Utama, I. D. C., & Winantu, A. (2013). Aplikasi Panduan Wisata Kabupaten Kulon Progo Berbasis Android dengan Menggunakan ADT Eclipse. *STMIK El Rahma*.

Wahyujati, D. M. (2017). *Implementasi Teknologi Firebase pada Aplikasi Pencarian Lokasi Service Kamera Berdasarkan Rating Berbasis Android*. STMIK AKAKOM Yogyakarta.

Wiratno, A. R., & Hastuti, K. (2017). Implementation of Firebase Realtime Database to Track BRT Trans Semarang. *Scientific Journal of Informatics*. https://doi.org/10.15294/sji.v4i2.10829