

Table of Contents

Abbreviations	2
1. Introduction	3
1.1. Background	3
1.2. Problem Statement.....	3
1.3. Scope	3
1.4. Overview of the document	3
2. System Architecture.....	4
2.1. System Overview.....	4
2.2. System Flow	5
3. System Engineering Overview	6
3.1. System Developmental Tools.....	6
4. System Model	7
4.1. Use Case Diagram	7
5. Functional and Non Functional Requirement.....	8
5.1. Functional Requirement	8
6. System Evolution:.....	9
6.1. Fundamental Assumptions for Hardware/Software:.....	9
7. System Requirement.....	10
7.1. System Requirement:.....	10

Abbreviations

- GPS: Global Positioning System
- API: Application Programming Interface
- IDE: Integrated Development Environment
- ADT: Android Developer Tools
- JVM: Java Virtual Machine
- JDK: Java Development Kit

1. Introduction

1.1. Background

There are numerous android application focusing on the navigation and information system. Google Map is a giant web dependent application that give users information regarding their position, bearing, and also information about places all over the world. Internet access is required at all times for these large scale application and they lack precise and adequate information about a particular area. There is a lot of potential in this field of navigation and Information system especially in context of our country.

1.2. Problem Statement

For the first time visitors of Kathmandu University, it is evident that the foremost problem is the lack of information and knowledge about the various departments and buildings of KU premises. People get confused as to where to go or where they currently are. We ourselves have faced this problem and upon inquiry with some newcomers, we conclude that there is a necessity for a means to gain information on the areas and buildings around KU. Hence, our group decided to make an App that can run offline and will enable users to move around KU with ease and get information about the buildings around KU. “KAPP” was the resulting app that we propose to create.

1.3. Scope

KAAP (KU App) is a navigation and information based android application that provides users with dynamic location feedback and information of Kathmandu University(KU) and its constituting buildings and departments. The application uses Google Maps and GPS system to enable users to navigate around KU area and gain information about their surrounding buildings. The users can ask for a route to a particular building and they will be shown a path of shortest distance to the inquired building.

1.4. Overview of the document

This document contains the detailed information about the app “KAPP”. This document is intended for the use of the users of the apps, future developers and also the client users who intend to use the app for their use. There is detailed information about the functional and non-functional requirements, system architecture, system evolution and use. This can also be considered as a manual for future change, modification and repairing/debugging of the app.

2. System Architecture

2.1. System Overview

The system consists of three components:

- **App:**
The App is the main component which can interact both with users and the admins.
- **User:**
Users can perform only the read-only functions of the application.
- **Admin:**
Admin can modify the content of the app. They can add/remove the content of the database through the app.

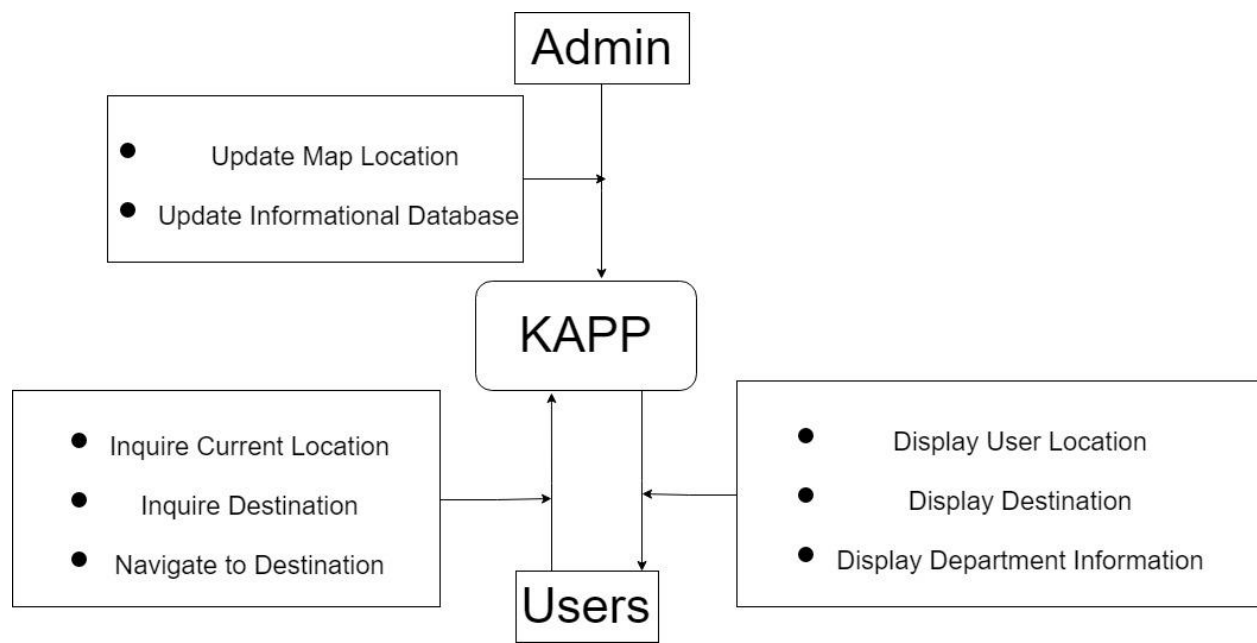


Figure 1: System Overview

2.2. System Flow

The following diagram shows how the system reacts to the user of the application. There is a separate method of handling the normal user and the Admin of the application.

3. System Engineering Overview

3.1. System Developmental Tools

The following are the development requirement for the app. Since the product in development is a software, we require other software tools to develop the app.

- **Android Studio:**
Android Studio is the official IDE for Google's Android OS, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse ADT as primary IDE for native Android application development.
- **Java Development Kit:**
The Java Development Kit (JDK) is an implementation of either one of the Java Platform, Standard Edition, Java Platform, Enterprise Edition, or Java Platform, Micro Edition platforms released by Oracle Corporation in the form of a binary product aimed at Java developers on Solaris, Linux, macOS or Windows. The JDK includes a private JVM and a few other resources to finish the development of a Java Application.
- **Android Emulator:**
Android Emulator is required to test and debug the application during the development phase. Smartphone can be used as a standalone android emulator by connecting it to android studio. Also android studio has a built in emulator which can emulate desired smartphone ROMs.
- **Google Maps API:**
Launched by Google in June 2005, the Google Maps API allows developers to integrate Google Maps into their application. For the development of this android app, the app developer has to obtain a developer's key from Google's developers website which gives a free license to use and modify the map on their application.

4. System Model

4.1. Use Case Diagram

The figure below shows how user and Admin handle the app. We can see

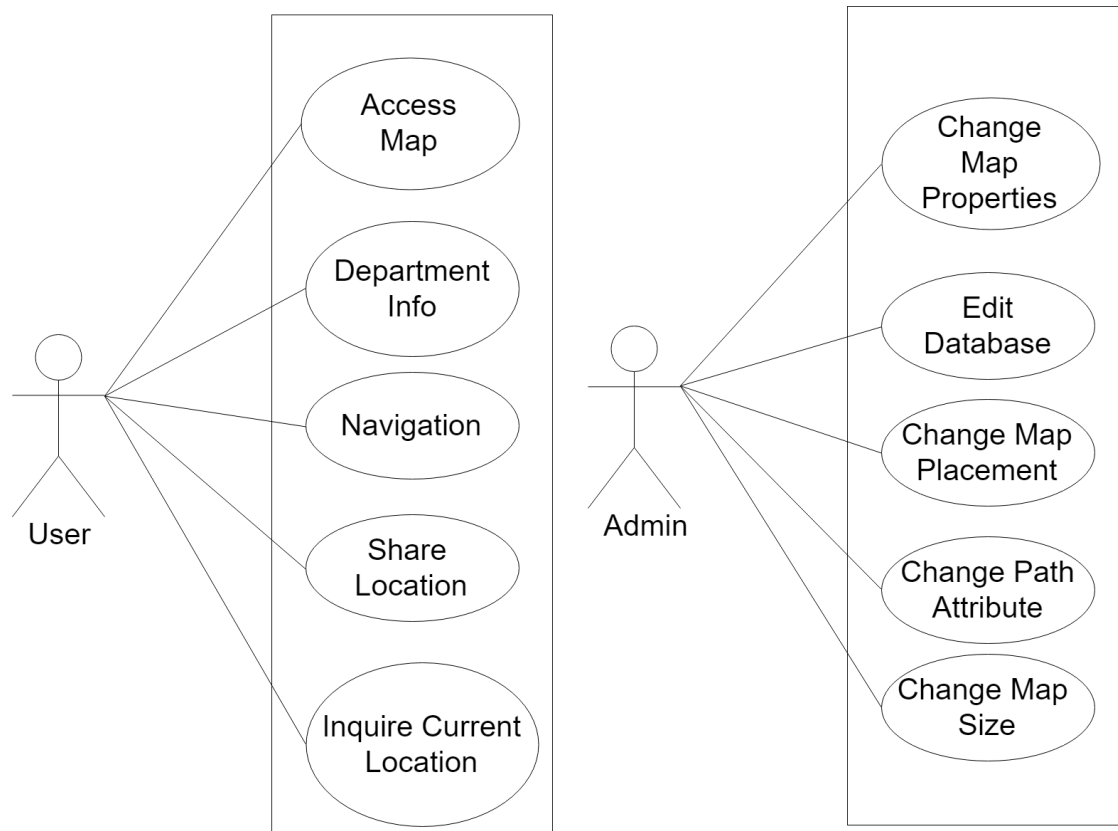


Figure 2: Use Case Diagram for User & Admin

5. Functional and Non Functional Requirement

5.1. Functional Requirement

The functional requirements describe the core functionality of the application. It gives the detail description on the feature and function of the application. The Functional Requirements Specification documents the operations and activities that a system must be able to perform. The major functional requirement of this application are as follows:

- The user can find all the block information of KU and excerpt in the home page and it is linked to more detailed information page.
- The user can see the detailed block information on respective block page, and navigate with direction, see indoor map, see top destination and faculty.
- The user is able to navigate the nearby blocks with respective to the user's current location

5.2. Detailed functional requirement

5.2.1. Common Requirements

6. System Evolution:

This section describes the fundamental assumptions on which our system is based, and any anticipated changes due to hardware evolution, changing user needs, and so on. This section is useful for system designers. It helps them avoid design decisions that would constrain likely future changes to the system.

6.1. Fundamental Assumptions for Hardware/Software:

- The application is based on Android OS and so the application is totally platform dependent and hence cannot be migrated to iOS or Windows Platform. The system will be updated for the updating versions of Android OS. The app will not run on Android OS below Android API level 19 (KitKat) version 4.4.
- The system is mainly software dependent. So as long as the application is run on Android device with GPS system, the application will not require and changes. The updates or changes on the hardware will not have any impact on the application.
- The Application will support a single user from a single application. Any number of users can use the application from the different smartphone at the same time.
- **Google Maps API use clause:**

7. System Requirement

7.1. System Requirement:

To run our application, the following are the system requirements:

- Android OS: Android API level 19 (**KitKat**) version 4.4 and above
- RAM: approx. 11 MB
- Dependencies:
 - Google Play Services-maps: 10.0.1
 - Google Play Services-location: 10.0.1
- Storage:
 - Package size: 3.10 MB
 - Installed File size: 8.85 MB
- Internet required for first time installation and use