**CS673 Software Engineering** 

**Team 1 - Hoopfinder**

**Software Design Document**

|  |  |  |  |
| --- | --- | --- | --- |
| Team Member | Role(s) | Signature | Date |
| Saloni Rawat | Team lead, Config leader |  |  |
| Mike Zhong | Back up team lead, Design and implementation leader |  |  |
| Jamie Smart | QA Leader |  |  |
| Sriram Doss | Requirements leader |  |  |
| Douyao Zhang | Security leader |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Revision history**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Author** | **Date** | **Change** |
| **01** | **Saloni Rawat** | **3 Oct 2019** | **Initial draft** |
| **02** | **Saloni Rawat** | **11 Oct 2019** | **Updated logo, introduction** |
| **03** | **Sriram Ramdoss** | **17 Oct 2019** | **Updated Database - Firebase** |

[Introduction](#_heading=h.gjdgxs)

[Software Architecture](#_heading=h.30j0zll)

[Design Patterns](#_heading=h.1fob9te)

[Key Algorithms](#_heading=h.3znysh7)

[Classes and Methods](#_heading=h.2et92p0)

[References](#_heading=h.tyjcwt)

[Glossary](#_heading=h.3dy6vkm)

# Introduction

*In this section, give an overview of this document, and also address the design goals of your software system.*

Hoopfinder is an android application that allows users to connect to other nearby interested players to organize events at nearby courts. More details on the project are available under the [SPPP](https://docs.google.com/document/d/1xOXwoxuaSoGwdceJBj4nmJrN1tKBkyNZ/edit)

The purpose of this document is to provide documentation to be used in the design and development of Hoopfinder. SDD - Software Design Document - is a design description of the system to allow for the development to proceed with an understanding what is the main expectation[[1]](#footnote-0).

The main design goals for the projects are:

1. Identify a database to securely store the user information as well as the location and court details. After an initial research on SQLite, the team decided to move forward with Firebase to store the user credentials. The data regarding the court location will also be saved on Firebase. We will also be storing the relations between users and between user and courts ( Which user follows which other user /court)
2. Easy Integration: Since this will be the first time that the team will be working on android development and its components, it is important that the integration of individual component developed by the team member is seamlessly and doesn’t lead to additional error and efforts.

# Software Architecture

*In this section, you will describe the decomposition of your software system, which include each component (which may be in terms of package or folder) and the relationship between components. You shall have a diagram to show the whole architecture, and class diagram for each component. The interface of each component and dependency between components should also be described. If any framework is used, it shall be defined here too. Database design should also be described if used.*

As mentioned earlier, the application will be built on android platform. Android is an open source, Linux-based software stack which can be used by a number of different devices[[2]](#footnote-1).

## Architecture Diagram

## Android Platform

The following diagram from Android developer, aims to elaborate on the major components of the platform.

**Figure 1.** The Android software stack.

## The app is built using Android Studio, where a new activity is created for each user story where user interaction is required.

**Authentication:**

In order to allow users to log into the application, we will be using Facebook API. This will allow users to log in using their facebook credentials and will reduce the effort required to keep the authentication information secure. The user information will be stored on Firebase database which is further explained below in the section.

## 

## Database - Firebase:

****

For storing personal information as location information, we will be using Firebase.

Firebase provides a realtime database and backend as a service. The service provides application developers an API that allows application data to be synchronized across clients and stored on Firebase

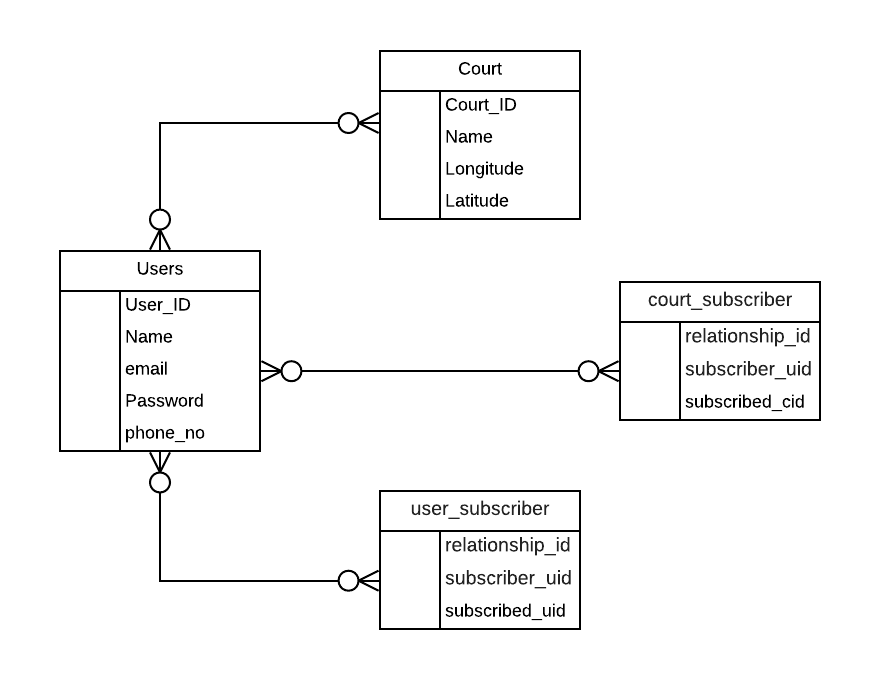
* Authentication - It can be handled both for username/email/password and for social authentication

Firebase Security Rules -It’s stand between your data and malicious users. You can write simple or complex rules that protect your app's data to the level of granularity that your specific app requires.

Google Analytics - It is a free app measurement solution that provides insight on app usage and user engagement.

Realtime Database - Firebase's original database. It's an efficient, low-latency solution for mobile apps that require synced states across clients in realtime.

Apart from authentication, we will be using the firebase for storing the following information as well.



## Packages

|  |  |  |
| --- | --- | --- |
| **Name** | **Purpose** | **Comments** |
| App | Contains   * Facebook login * User location information. * User information |  |
| Notifications | Classes for push notifications | 10/17: Needs to be merged to apps. |

## Use Cases Diagram

## Class Diagram

# Design Patterns

In this section, you shall describe any design patterns used in your software system.

# Key Algorithms

In this section, you shall describe any key algorithms used in your software system, either in terms of pseudocode or flowchart.

# Classes and Methods

This part can be a reference to automatic generated document for all classes and methods.

# References

<http://robotics.ee.uwa.edu.au/courses/design/examples/example_design.pdf>

<https://developer.android.com/topic/libraries/support-library>

<https://arxiv.org/ftp/arxiv/papers/1005/1005.0595.pdf>

# Glossary

1. <http://robotics.ee.uwa.edu.au/courses/design/examples/example_design.pdf> [↑](#footnote-ref-0)
2. <https://developer.android.com/guide/platform> [↑](#footnote-ref-1)