**Pocket Advisor**

**Software Requirements Specification Document**

Andreas Adolfsson

Michael Barrera

Abraheem Omari

1. **Introduction**
2. **Purpose**

The purpose of this document is to give a detailed description of the requirements for the Pocket Advisor Application. This document will cover each of the systems intended features, software, and various technical dependencies. It is intended to be viewed by the project members that will implement and verify the system.

1. **Scope of the System Specified**

This software is a mobile Android application that will allow FSU students to keep track of their college courses. The application will allow for adding of courses and course details (including pre-requisites, co-requisites, etc.) to the application. The application will then be able to quickly and accurately generate a visual representation of an optimized career plan.

This visual realization of the students career plan clarifies which classes the student should take during in relation to the semesters that they should be taken to graduate as quickly as possible. The goal of this application is that come registration time, students will be able to quickly and conveniently determine which courses they should sign up for. Additionally, this application aims to keep the student more informed of their actual degree progress.

Should the student not be satisfied with their generated schedule, mechanisms will be in place to allow the student to edit the schedule in real time.

Additionally, there shall be the ability to choose a pre-set major with all of the required classes automatically added to the student’s schedule.

1. **Definitions, Acronyms, and Abbreviations**

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **Class** | **A course in the system** |
| **Schedule** | **A schedule comprised of Classes** |
| **Height** | **The longest chain of requisites that a course has** |
| **Relevance** | **The amount of other courses in the schedule that requires the course as a requisite.** |

1. **References to Supporting Document**

IEEE. IEEE Std 830-1984 IEEE Guide to Software Requirements Specifications. IEEE

Computer Society, 1984.

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.* IEEE Computer Society, 1998.

1. **Overview of rest of SRS**

The next chapter, the General Description section, of this document gives an overview of the general functionality of the system. It describes general constraints, dependencies and assumptions the system relies on. In addition, it also describes the products perspective to its environment, audience and the functions it performs.

The third chapter covers the functional requirements of the system. This chapter will list the functions that the system shall do.

The fourth chapter covers the non-functional requirements of the system. This follows the same format as the previous chapter; however it will list the criteria that the operations of the system can be judged on rather than its behavior.

The fifth chapter offers a class diagram for our system that details its classes and their attributes. Also included is a brief description of the system as it is depicted.

The sixth chapter illustrates through use-case and sequence diagrams the system to be developed. It is followed by a brief description of these diagrams.

1. **Description**
2. **Product Perspective**

This system operates within the bounds of the Android operating system. Otherwise, it is an entirely independent and stand-alone application. Users interact with it through the touch screen on their devices. Data will be stored in a local SQLite database. As of now it is not intended to have any connection to the internet or remote services.

1. **Product Functions**

Using this software users shall be able to:

* Add courses to degree plan
* Delete courses from degree plan
* Modify course details such as pre-requisites
* Generate degree schedule
* View degree schedule organized by semesters
* Mark courses as completed
* Enter final grade for completed courses
* View percentage of degree completed
* View GPA as calculated by provided grades
* Manually modify schedule to best suit personal needs
* Choose pre-set major and auto-populate schedule
* Save all schedules, courses, grades, etc. in a local database

1. **User Characteristics**

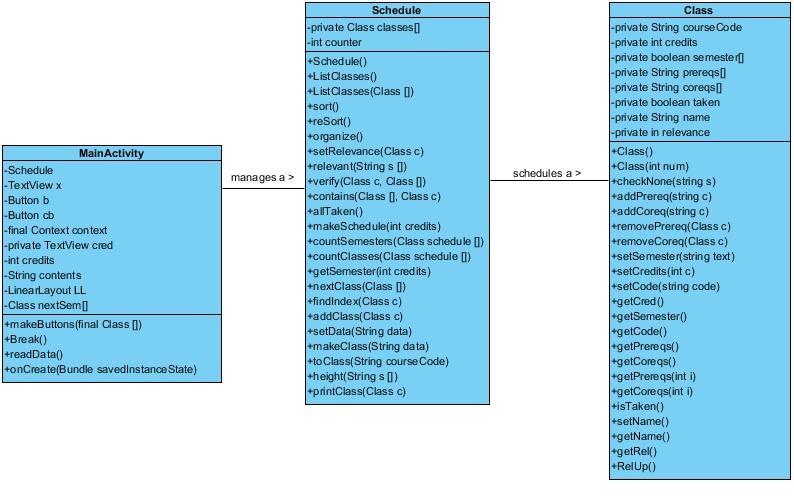
Users of this software are college students likely attending Florida State University. They are primarily between the ages of 18-24. They may be in significant debt. They own at least one Android device and are likely familiar with apps and are competent in similarly complex technologies. They should be able to fully utilize this software with minimal instruction.

1. **General Constraints**

The software must be designed on the Android platform and must be capable of running on average to sub-average hardware. The Android platform will require the software to be natively written in Java in conjunction with the Android framework. What’s more, the software will be limited to the English language and English speaking/reading users.

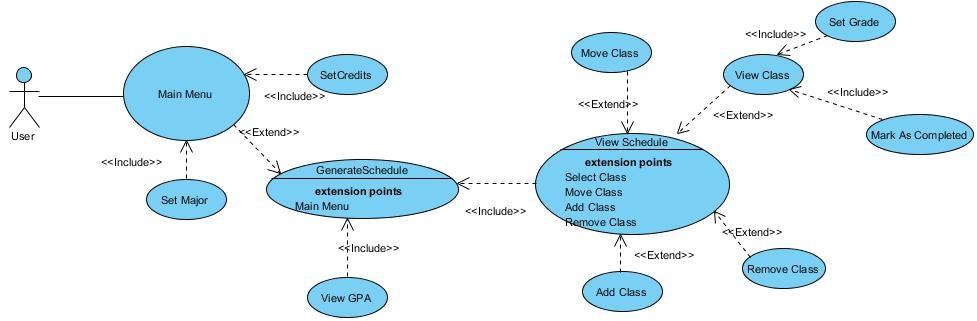
1. **Assumption, and Dependences**

It is assumed that the university will continue to use the same GPA scale. It is also assumed that the university will continue using the Fall/Spring semester model.

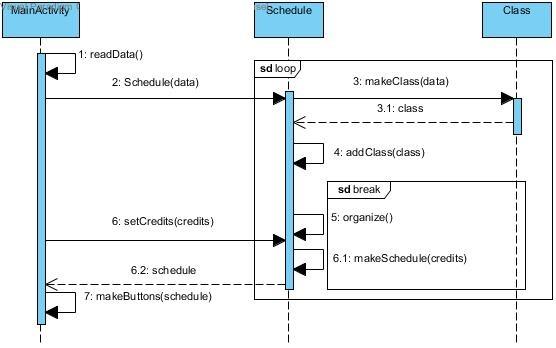
1. **Functional Requirements**
2. A user shall be able to add a course to their schedule
3. A user shall be able to view a list of courses relevant to their major
4. A user shall be able to input the final grade for a completed course
5. A user shall be able to view their GPA based off the grades of completed courses
6. A user shall be able to input the minimum amount of credits to take per semester
7. The system shall be able to automatically generate a semester schedule based off minimum credits per semester and incomplete courses
8. A user shall not be able to take a course if prerequisite courses have not been taken
9. A user shall be able to mark classes as completed
10. The system shall prevent the same class from being taken more than once per semester
11. **Non-functional Requirements**
12. The system should run on a mobile Android environment
13. The data should come from an FSU source
14. The system should be reusable
15. The language should be uniform
16. **System Architecture**

The foundation of the system is based on these three classes. The MainActivity calls upon the Schedule class via the makeSchedule() function. This function calls upon private methods in the Schedule class and ultimately returns an array of Class objects representing the students generated schedule for their degree. The MainActivity then generates the visual representation of this schedule.

1. **System Model**



The user has access to the main menu where they can set the credits they want to take as well as their major if it is available. They can also add/modify classes, mark them as completed, set their grades, view their GPA, view their schedule, and regenerate it as needed.



1. **Appendices**
   1. **Data dictionary** 
      1. Actor Descriptions
         1. User: The person using the application
2. Use Case Descriptions
   1. The use case shows what the user can do upon entering the application. Actions range from selecting a major, generating a schedule, and viewing classes in the schedule
3. Class Descriptions
   1. MainActivity: The main thread of the program. Handles the reading of class data and the creation of a schedule, and the displaying of the schedule.
   2. Schedule: Creates classes and arranges them into a schedule starting from the first semester to the last semester of college.
   3. Class: Holds all class information such as course number, credits, the user’s grade, course name, the requisites for the course, and the semester it is offered.