Software Requirements Specification

for

GradeUP™

Version 1.0 approved

Prepared by <author(s)>

<Organization>

<Date created>

Table of Contents

Table of Contents ii

Revision History iii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Project Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Features 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

3. System Features 4

3.1 System Feature 1 4

3.2 System Feature 2 (and so on) 4

4. External Interface Requirements 5

4.1 User Interfaces 5

4.2 Hardware Interfaces 5

4.3 Software Interfaces 5

4.4 Communications Interfaces 5

5. Other Nonfunctional Requirements 6

5.1 Performance Requirements 6

5.2 Safety Requirements 6

5.3 Security Requirements 6

5.4 Software Quality Attributes 6

6. Key Milestones 7

7. Key Resource Requirements 8

8. Other Requirements 9

Appendix A: Glossary 9

Appendix B: Analysis Models 9

Appendix C: Issues List 9

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

The purpose of this document is to present a detailed description of the GradeUP application. It explains the purpose and functionalities of the application, its user interfaces and the various requirements for successful operation of the application. This document is intended for both the stakeholders and the developers of the system.

## Document Conventions

The format of this document is simple. Bold face and indentation is used on general topics and or specific points of interest with diagrammatic visualization at few places. The remainder of the document has been written using the standard font – Arial.

## Intended Audience and Reading Suggestions

This document is targeted to the stakeholders, client, developers and testers for the project.

Readers interested in a brief overview of the product can refer to section 1 – Introduction, as well as section 2 – Overall Description, which provide a brief overview of each aspect of the project as a whole.

## Project Scope

The GradeUP android application is being designed to ease the user’s effort in studies. It is targeted for anyone who wants to collaborate & share knowledge efficiently by the means of study groups but are facing scheduling problems due to conflicting timetables. The application helps all users in a group find a common time slot during which they are all free to meet for study. It helps the user in finding friends who are free to help with the specific subjects and form study groups to simplify the process of studying. It also aims at providing a forum for general subject related discussions. The application can be used to effectively manage the group meetings. This will reduce the effort required in getting help for completing a course or subject and would ideally help improve the grade of a student who uses it or make it easy for anyone trying to learn any subject. It will help the user finish their coursework in an organized manner. Thus, the application has universal scope and can be used by anyone. The long term vision is to make everyone’s student life easier and help them get organized.

## References

IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.

IEEE Computer Society, 1998.

# Overall Description

## Product Perspective

The most effective to way to study a topic amongst a number of people is to form study groups. However the most common problems people face while forming a study group is scheduling and setting up an agenda. GradeUP seeks to address these issues. The application will allow users to synchronize schedules with other group members and indicate time slots during which all members are free. The members will be able schedule meetings at these times and set an agenda for the meeting. The group members will be able to discuss topics and queries on a discussion forum within the application. As an added feature the users will get a reminder of the meeting 15 minutes before the allotted time. Also during the scheduled meeting, social media notifications of the group members will be blocked, so that the members can focus on the meeting.

## Product Features

The application has the following feature which will be discussed in detail in the sections below.

* Allows the user to form study groups for specific subjects
* Calendar to display when everyone in the group is available and schedule meetings at those times
* Setup agenda for every meeting and track progress of individual members
* SMS and email alerts for meetings
* Start discussion forums and post queries and doubts related to subjects
* Block social media notifications while group is in sessions

## User Classes and Characteristics

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the favored user classes from those who are less important to satisfy.>

## Operating Environment

GradeUP will be a mobile based application developed on the Android OS platform. It will be supported by Android versions 4.0 and above. The database will be SQLite. XML will be used to encode the data and JAVA will form the bridge between the front and back end.

## Design and Implementation Constraints

The GradeUP design and UI will have to be intuitive and not have cumbersome UI. The application should be quick and responsive. The applications must also be accurate in suggesting meeting time and give out timely reminders of an upcoming meeting.

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).>

## User Documentation

The application will be designed to be intuitive to issue. Nevertheless a ‘Help’ section will be provided which describes the way in which the application and its features are to be used.

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

## Assumptions and Dependencies

The application will involve extensive use of Android and SQLite technologies. The team has little experience in these technologies so a steep learning curve can be expected to cause delays.

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

# System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

## Register/Login/Logout

**3.1.1 Description and Priority**

The Register/Login/Logout feature allows the user to Register/login/logout of the Application. The user will have to enter their first name, last name, email ID, registered courses and contact number.

**3.1.2 Stimulus/Response Sequences**

* When the user first opens the application he/she will be greeted with a Register/ Login screen
* A first time user will be asked to register for the application.
* He/She will be asked to enter First Name, Last Name, Email ID, Password, Contact number and Courses Registered
* A returning user will be asked to login by entering his/her email ID and Password
* If the user successfully logins he/she will be redirected to their profile page.
* If the wrong Email ID or Password is entered a message saying invalid username or password will be shown
* A Logout button will appear on top of the screen. When the logout button is clicked the user is logged out of the application and redirected to the login page.

### 3.1.3 Functional Requirements

REQ-1: The Email ID and Password should be unique and validated before allowing the user to register

REQ-2: The data of the user will have to be encrypted and stored in the database.

REQ-3: The password should not be visible as characters but instead should appear as ‘\*’

REQ-4: The main home screen should be loaded only if the username/password combination is correct, otherwise the error message should appear

REQ-5: Once the user is logged out the user should not be able to go back into the application without logging on again

## Create a group

**3.2.1 Description and Priority**

This feature allows the user to create a new group. When a user creates a group he/she automatically becomes the admin for that group

**3.2.2 Stimulus/Response Sequences**

* User clicks on ‘create a group’ button- The user is prompted to enter a group name and once he/she clicks ‘submit’ a new group is created. This user becomes the Admin user.
* Admin user can add members to the group

### 3.2.3 Functional Requirements

REQ-1: The new group should be saved in the database and should be available whenever the user views list of groups created by him/her.

REQ-2: There should be no duplicate group name for each particular user i.e the same user cannot have 2 groups of the same name

## Join/Leave a group

**3.3.1 Description and Priority**

This feature allows the user to join an existing group. A user can also leave the group that they are a part of

**3.3.2 Stimulus/Response Sequences**

* User goes through a list of groups and clicks on ‘join a group’ button for a particular group. This sends a request to the admin. If the admin accepts the users request the user is added to the group.
* User can leave the group by clicking ‘leave group’ button

### 3.3.3 Functional Requirements

REQ-1: The user data should be added to the members list of the group in the DB and vice versa

REQ-2: The user should be able to post/start discussions in the group once he/she has joined a group

REQ-3: On leaving a group all related data to the particular group should be removed from the DB for that particular user and should not show up again on that user’s list of groups

## View ‘My Dashboard’

**3.4.1 Description and Priority**

The dashboard screen is a central location where the user can see his/her schedule for the day, the list of groups the user is a part of and the general discussion portal.

**3.4.2 Stimulus/Response Sequences**

* The user is redirected to the dashboard on login.
* The user will be able to see his/her schedule.
* The user will be able to scroll through the list of groups that he/she is a part of
* The user clicks on a particular group name- he/she is directed to the group page and the data related to the group is loaded

### 3.4.3 Functional Requirements

REQ-1: The user data should be able to see all the groups that he/she is a part of

REQ-2: The user should be able to navigate to the particular group by clicking on the group name

REQ-3: The groups created by user and the groups that he/she is just a part of should be differentiated in the list by color/symbols

REQ-4: The user will be able to view the general discussion portal and post on the portal

## View list of groups

**3.5.1 Description and Priority**

This shows the groups that exist within the application that the user isn’t a part of. The user

can join the group by clicking on ‘join group’ as mentioned in section 3.3

**3.5.2 Stimulus/Response Sequences**

* The user will click on the sidebar to open the options menu and then click on the join group option.
* The user will be able to scroll through the list of groups that exist in the application
* The user can join any group by clicking on join group option of a group

### 3.5.3 Functional Requirements

REQ-1: The user should be able to see the groups that he/she is a not a part of

REQ-2: The user should be able to navigate to the particular group by clicking on the group name

REQ-3: The user should be able to join any group on clicking the ‘join group’ button

## Free time in schedule

**3.6.1 Description and Priority**

The user on entering a group should be able to see the combined schedules of the group and the free time available to all

**3.6.2 Stimulus/Response Sequences**

* The user will be able to see the combined schedules of his/her group members on entering the group and clicking on the calendar option

### 3.6.3 Functional Requirements

REQ-1: The user should be able to see his/her schedule for the week along with the study group meetings of his/her group

REQ-2: The application should calculate which time slots are free for the users in a group

## Schedule/Cancel meetings

**3.7.1 Description and Priority**

The user should be able to schedule a meeting with a particular study group based in the common free time available to all.

**3.7.2 Stimulus/Response Sequences**

* User on entering the group will be able to the common free time available to all. The user should be able to select a free time slot for a meeting
* When a user select a time slot for the meeting that slot should appeared blocked on the users calendar.

### 3.7.3 Functional Requirements

REQ-1: The user should be able to select a time slot and reserve it for a meeting time

REQ-2: The booked time slot should appear as reserved on the other groups the member might be in.

## Setup Agenda for meetings

**3.8.1 Description and Priority**

The user should be able to setup an agenda for the meeting

**3.8.2 Stimulus/Response Sequences**

The user will receive SMS and email alerts about the meetings and the tasks which he/she has to complete before the meeting commences.

### 3.8.3 Functional Requirements

REQ-1: The agenda should be visible to all members in the group

## Start/comment discussion within the group

**3.9.1 Description and Priority**

The user should be able to start a discussion thread, where he/she can post doubts and queries within a group. Other members in the group should be able to respond to the queries.

**3.9.2 Stimulus/Response Sequences**

* User on entering the group will be able post queries or start discussion threads. The queries will be entered in textbox and posted by clicking a ‘Submit’.
* Other members in the group should be able to respond to the post using a textbox and a reply button.

### 3.9.3 Functional Requirements

REQ-1: The user should be able to post a query on the group via a textbox. Once posted the query will be stored on database and loaded onto the group members phones.

REQ-2: The other member will post their reply via text boxes.

REQ-3: The discussions thread will be stored on Server.

## Start/comment on public discussion thread

**3.10.1 Description and Priority**

The user should be able to start a public discussion thread where he/she would be able to post any doubt or queries. Other users should be able to see these threads and be able to comment on them.

**3.10.2 Stimulus/Response Sequences**

* User will be able post queries or start discussion threads on the public forum. The queries will be entered in textbox and posted by clicking a ‘Submit’.
* Any user should be able to respond to the post using a textbox and a reply button.

### 3.10.3 Functional Requirements

REQ-1: The user should be able to post a query via a textbox. This thread would be visible to everyone on the public discussion forum

REQ-2: Once posted, other users should be able to reply to these queries using textbox.

## SMS and Email Reminder

**3.11.1 Description and Priority**

The user should receive a SMS and email reminder on their registered phone number and email ID 15 minutes before the start of the meeting

**3.11.2 Stimulus/Response Sequences**

* The user enters his phone and email ID at the time of registration.
* The application will give a SMS and email reminder 15 minutes before the meeting.

### 3.11.3 Functional Requirements

REQ-1: The application will send an automated SMS and email shortly before the meeting time

## Block Social Media Notifications

**3.12.1 Description and Priority**

For the duration of the study group meeting social media notifications should be blocked to prevent the group members from getting distracted from studying

**3.12.2 Stimulus/Response Sequences**

* When a meeting is scheduled the social media notifications will automatically

Shut off for the duration of the meeting

### 3.12.3 Functional Requirements

REQ-1: The application will have to access notification settings of the mobile to switch of the social media notification at the appropriate time

# External Interface Requirements

## User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

## Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

## Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

SQLite

Java

## Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

# Other Nonfunctional Requirements

## Performance Requirements

All queries are predefined and do not involve any large amounts of data, thus synchronization of data with database server and any other updates should happen very quickly. Switching between screens will happen seamlessly as application will be using the cache memory.

## Safety Requirements

N/A

## Security Requirements

N/A

## Software Quality Attributes

The application will be compatible with any mobile device running on Android OS v4.2 or above bearing a user friendly and visually aesthetic user interface for ease of use and for navigating between screens.

To maintain application robustness, thorough testing will be done to ensure zero application crash instances.

Reliability and correctness of the scheduler algorithm will be maintained such that it provides optimum available meeting times for the group members.

In case of an internet connection outage, the application features will be partially available but no new data updates will happen during this time.

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

# Key Milestones

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Milestone** | **Target Completion Date** | **Comments** |
| 1. | Sprint 1 | 10/05/2015 |  |
| 2. | Sprint 2 | 11/02/2015 |  |
| 3. | Sprint 3(Final Sprint) | 11/30/2015 |  |

# Key Resource Requirements

*<Identify the skill or expertise required for each major project activity. Identify any appropriate internal staff resources available. Indicate whether a resource only can commit part-time in the comments. If no internal resources are available, indicate the outside sources needed for the required skills. You may need more than one outside resource.>*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Major Project Activities | Skill/Expertise Required | Internal  Resource | External Resource | Issues/Constraints |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: Issues List

< This is a dynamic list of the open requirements issues that remain to be resolved, including TBDs, pending decisions, information that is needed, conflicts awaiting resolution, and the like.>