American International University-Bangladesh (AIUB)

Department of Computer Science

Faculty of Science & Technology (FST)

GPS-Based Reminder

A Software Requirement Engineering Project Submitted

By

Semester: Spring_22_23		**Section: [F]**	**Group Number: 5**	
SN	Student Name	Student ID	Contribution (CO1+CO2)	Individual Marks
34	PAUL, PULOK KANTI	20-43502-1		
38	RAHMAN, MINHAJUR	20-44166-2		
39	TONNY, TANIA YEASMIN	20-44168-2		
40	KAMRUZZAMAN, MD.	20-44222-3		

The project will be Evaluated for the following Course Outcomes

Evaluation Criteria	Total Marks (50)	
Introduction, Format, Submission, Defense	[10 Marks]	
System Overall Description & Functional Requirements	[10 Marks]	
System Quality Attributes and Project Requirements	[10 Marks]	
UML and E-R Diagram with Data Dictionary	[10 Marks]	
UI/UX Prototyping	[10 Marks]	

Software Requirements Specification

for

< GPS-Based Reminder >

SRS_ GPS-Based Reminder_V_1.0

Prepared by < TONNY, TANIA YEASMIN, RAHMAN, MINHAJUR , KAMRUZZAMAN, MD. , PAUL, PULOK KANTI >

< American International University-Bangladesh (AIUB)>

<April 29, 2023>

Table of Contents

Revision History

Name	Date	Reason for Changes	Version
N/A	N/A	N/A	N/A

1. Introduction

1.1 Purpose

This software requirements specification (SRS) document is created to specify the software requirements for a GPS-based reminder system. The product being specified in this document is the initial release of the software.

The scope of this SRS is to describe the requirements for the GPS-based reminder system, including the features, functions, and constraints of the system. This SRS covers the entire system and its subsystems, including the software application and database.

The GPS-based reminder system is a software application allowing users to set reminders based on their location. The system uses GPS technology to determine the user's current location and notifies them of reminders when they enter or exit a specified geographic area. The software application will be available on mobile devices, such as smartphones and tablets, and will be compatible with major operating systems.

The purpose of the GPS-based reminder system is to provide users with a convenient and efficient way to manage their tasks and reminders. The system will enable users to set reminders for various tasks, such as appointments, meetings, and shopping lists, and receive notifications when they are near the designated location. The system will also allow users to customize reminder settings, such as the reminder frequency, notification sound, and distance from the location.
The GPS-based reminder system is aligned with corporate goals and business strategies to provide innovative and useful software solutions to users. The software aims to enhance productivity and reduce stress by eliminating the need for manual reminders and increasing the efficiency of task management.

1.2 Document Conventions

The following standards and typographical conventions have been followed when writing this Software Requirements Specification (SRS) document for the GPS-based reminder system:

- **Font**: The font used throughout the document is Times New Roman with a font size of 12 points.
- **Headings**: Major headings are formatted in bold letters, and subheadings are in bold sentence case.
- **Prioritization**: Each requirement statement in the SRS document will have its own priority level, as specified by the customer. The priority level of the higher-level requirement is not assumed to be inherited by detailed requirements.
- **Use of Bold**: Bold font is used to highlight important terms and concepts, such as system components, user interface elements, and significant system functions.
- **Use of Italics**: Italics are used to denote references to other documents, such as a separate vision and scope document or external standards.

- **Use of Underline**: Underlining is used to denote hyperlinks to external resources or related documents.
- **Numbering**: Each requirement is numbered in a hierarchical structure. The numbering system follows the format: Major Requirement. Minor Requirement. Sub-Minor Requirement.
- **Traceability**: The traceability matrix will be used to trace each requirement to its origin and ensure that all requirements are covered.

1.3 Intended Audience and Reading Suggestions

The following types of readers are the intended audience for this Software Requirements Specification (SRS) document for the GPS-based reminder system:

1. **Developers**: Developers will use this document as a reference to **develop the software** for the GPS-based reminder system. They will use this document to understand the software requirements and develop the software that meets these requirements.

2. **Project Managers**: Project managers will use this document to **understand** the **scope** of the project, estimate the resources required for the project, and track the progress of the project.

3. **Testers**: Testers will use this document **to develop test cases** that verify that the software meets the requirements specified in the document.

4. **Documentation Writers**: Documentation writers will use this document **to develop user manuals**, help files, and other user-facing documentation for the software.

5. **Users**: Users will use this document to **understand the features and functions** of the GPS-based reminder system and its intended purpose.

1.4 References

- IEEE Standards Association. IEEE Recommended Practice for Software Requirements Specifications. (IEEE Std 830-1998). Available at https://ieeexplore.ieee.org/document/720574
- Techopedia. Software Requirements Specification (SRS). Available at https://www.techopedia.com/definition/13925/software-requirements-specification-srs
- ReQtest. How to Write a Software Requirements Specification (SRS) Document. Available at https://reqtest.com/requirements-blog/how-to-write-a-software-requirements-specification-srs-document/
- Software Engineering Institute. Writing Good Software Requirements. Available at https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=508184
- University of York. Writing a Software Requirements Specification. Available at https://www.cs.york.ac.uk/ftpdir/reports/UCS/ucsy11.pdf
- Guru99. What is Requirements Traceability Matrix (RTM) in Testing? Available at https://www.guru99.com/traceability-matrix.html

2. Overall Description

2.1 Product Perspective

The GPS-based reminder system is an overall system and a replacement for certain existing systems that will help users to create and manage reminders based on their current or specific location. The system will interact with the GPS of the user's mobile device to track their location and provide reminders when they enter or exit a specific geographic area. This system will also provide a simple and intuitive interface for users to create, edit, and delete reminders, which will be stored in a local database on the device.

Interoperability: The GPS-based reminder system will be designed to be interoperable with other mobile applications on the user's device. It will be able to import and export data to and from other applications, making it easy for users to share their reminders with others. Additionally, the system will support integration with third-party services, such as Google Maps or other location-based services.

Financial Benefit: The GPS-based reminder system will have a significant financial benefit for users who need to remember important events, such as appointments or meetings, when they are on the go. It will save users time and money by providing them with reminders when they need them, without the need to manually enter them into a calendar or other application.

Maintenance Cost: The GPS-based reminder system will have low maintenance costs, as it will be designed to work on standard mobile devices and will not require any special hardware or software to operate. Updates and bug fixes will be provided through the app stores where the application is available, and users will be able to download and install them as needed.

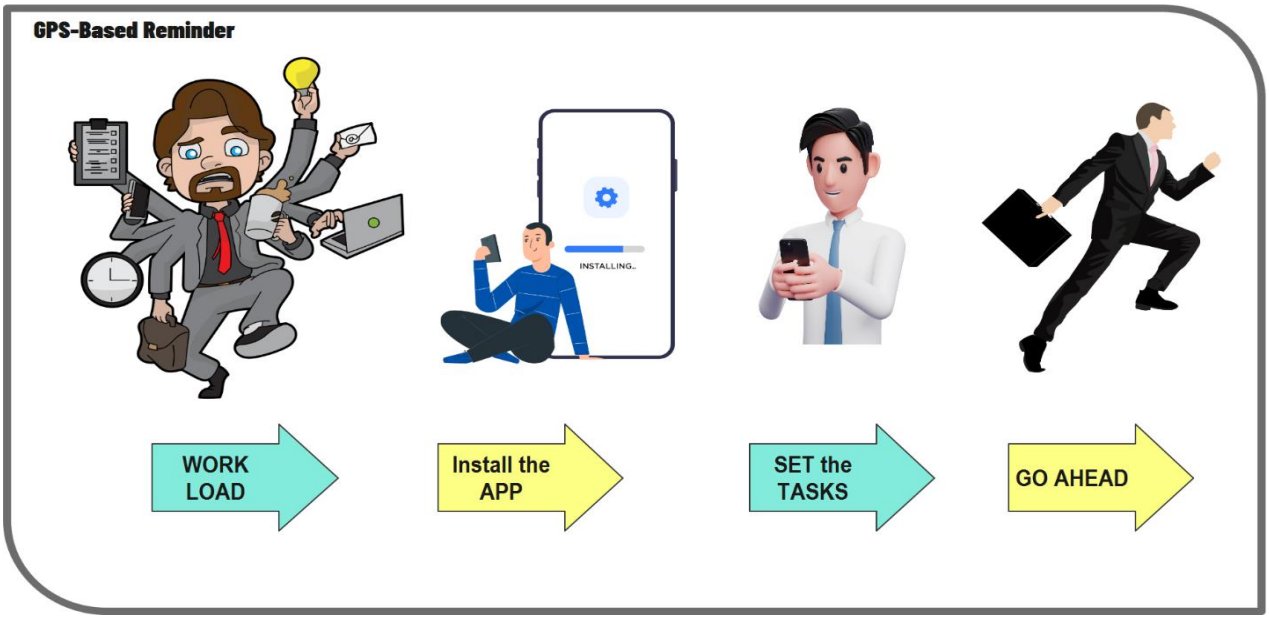


*Fig. **1**: Components of the overall system*

2.2 Product Functions

1. Registration:

- Allow users to register with personal information and a verified Google account.
- Verify user through a random code sent to email.
- Set a unique username and strong password.
- Display login page upon successful registration.

2. Set goals:

- Allow users to input information about tasks and their location.
- Enable users to attach a time if there is any offline possibility.
- Allow users to add collaborators via email if desired.
- Enable users to skip and reset tasks.
- Allow users to mark and display completed tasks.

3. Add socially:

- Display all users of the application from contact and social sites databases.
- Allow users to add friends externally through an invitation link.
- Show up-to-date rank by completed tasks.
- Allow users to share specific moments of their choice.

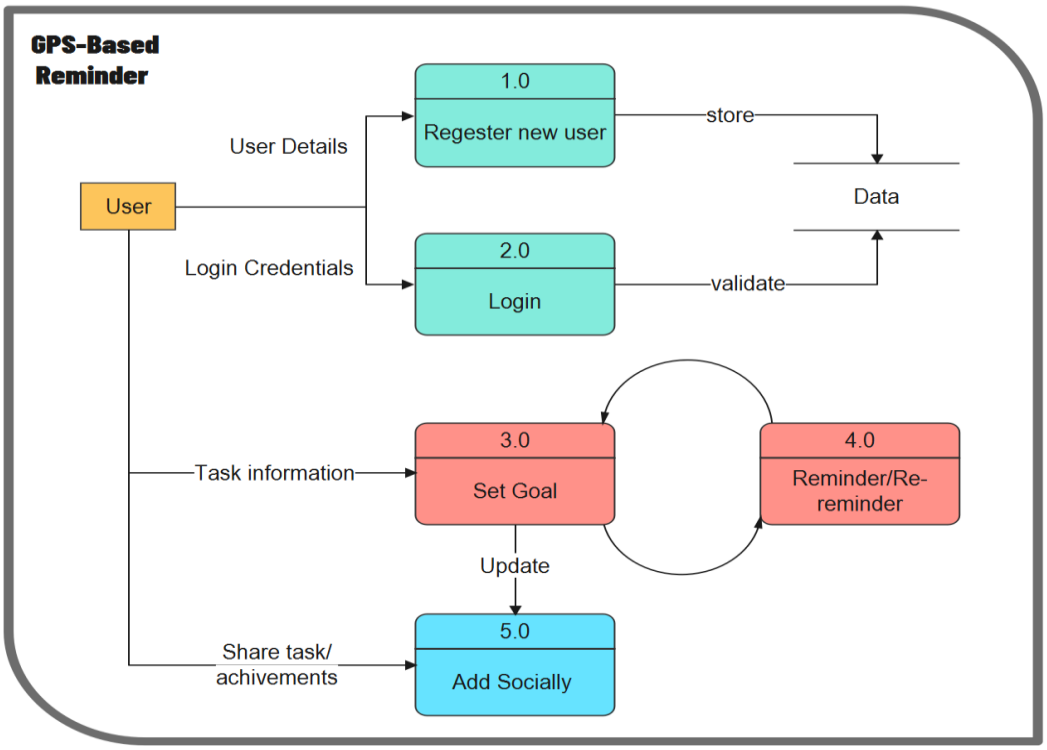


*Fig. **2**: Data flow diagram to show major groups of related Functional requirements and how they relate.*

2.3 User Classes and Characteristics

User Classes and Characteristics of GPS-Based Reminder System:

1. **General Users:**

- They are individuals who want to use the system for personal reminders.
- They may have varying levels of technical expertise and experience with location-based services.
- They are concerned with the usability and convenience of the system.

2. **Business Users:**

- They are individuals or organizations that want to use the system for business-related reminders and tasks.
- They may require additional features such as team collaboration and task assignment.
- They may have higher security and privacy requirements for their data.

3. **Technical Users:**

- They are individuals who have advanced technical expertise and may use the system for developing other applications.
- They require access to the system's API and technical documentation.

4. **Administrator:**

- They are individuals who manage the system and have access to sensitive information.
- They require advanced security and authentication features to ensure the privacy and security of user data.

The most important user class for this product is the general users, as they are the primary target audience, and the success of the product depends on their satisfaction with the system. The other user classes, while important, are secondary in importance and can be developed in later iterations of the system.

2.4 Operating Environment

The GPS-based reminder system will operate on **mobile devices** running on either **iOS** or **Android operating** systems. The mobile devices must have a GPS receiver and internet connectivity. The minimum operating system requirements for the mobile application are iOS 12 and Android 6.0 (Marshmallow) or above. The system does not require any specific hardware components beyond those included in standard mobile devices.

The backend server will require a web server, such as Apache or Nginx, running on a Linux operating system. The server must have adequate processing power, memory, and storage to handle the expected number of concurrent users and their associated data.

The system will interface with external services, such as mapping and geolocation services, to determine the user's location and provide accurate reminders. These services will need to be accessible through the internet and must be compatible with the system's API.

The GPS-based reminder system should be compatible and able to seamlessly integrate with other mobile applications and services, such as messaging and email apps, to effectively send reminders and notifications to users. Furthermore, it must also coexist with Google Maps to access location information, enabling the system to set location-based reminders.

2.5 Design and Implementation Constraints

- **Platform constraints**: The software should be designed and implemented in such a way that it can run smoothly on both Android and iOS platforms.

- **Integration with other systems:** The software must integrate with the GPS system on the user's device and also communicate with the server for synchronization with other devices

- **Security constraints:** The software must ensure data confidentiality and integrity by implementing proper encryption algorithms and access control policies.

- **User interface constraints:** The software must have a simple and intuitive user interface that is easy to use for all classes of users.

- **Performance constraints:** The software must be designed to work efficiently even on devices with limited processing power and memory. It must not have any significant impact on the battery life of the device.

- **Compliance/Permission constraints:** The software must comply with all relevant laws and regulations, such as data protection and privacy laws.

2.6 User Documentation

This user manual provides detailed instructions for using the system, including step-by-step guides below:

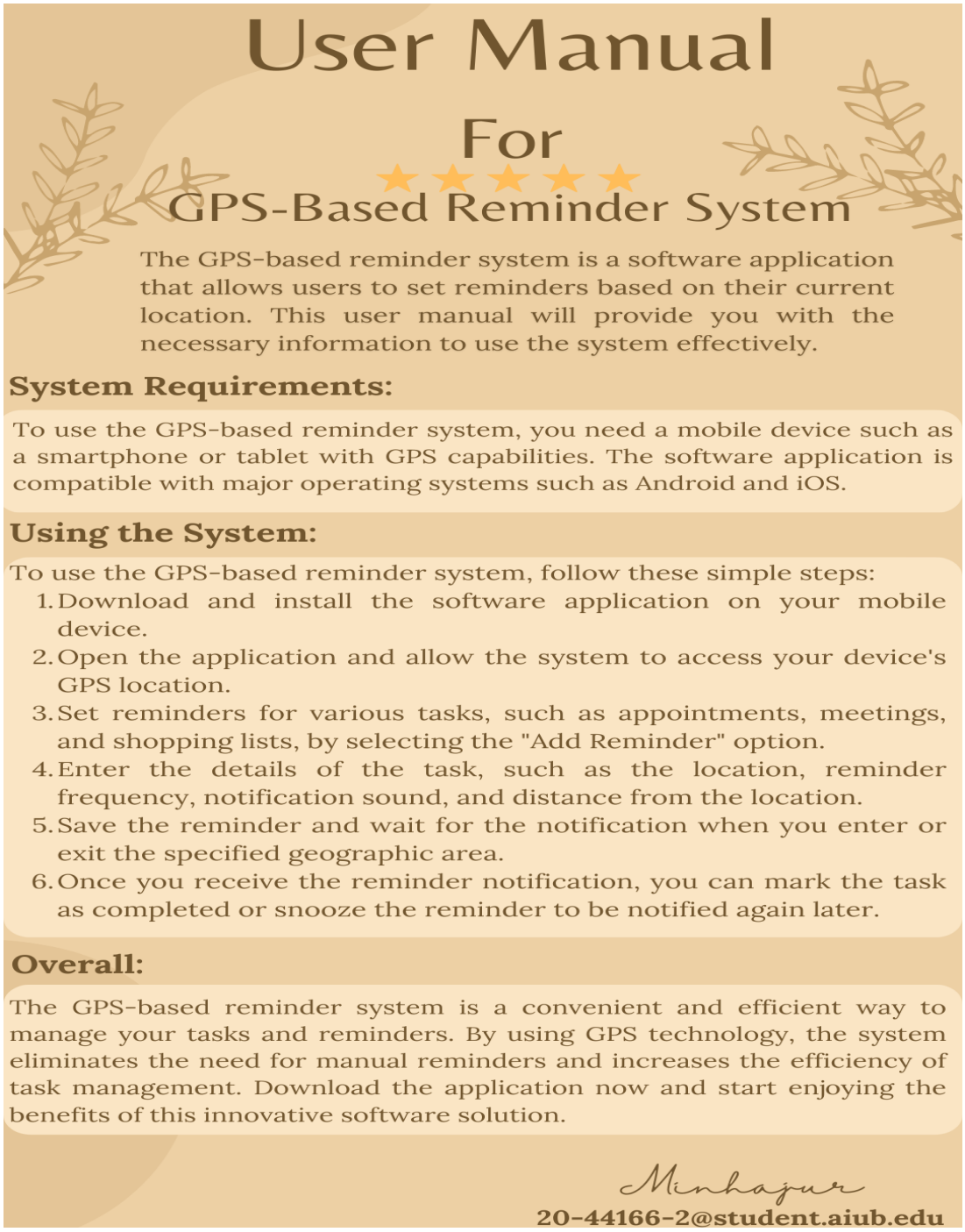


*Fig. **3**: User manual*

3. System Requirements

3.1 System Features

1. Registration

Functional Requirement.

1.1 The software shall allow users to start registration with their Personal Information (Name, DOB, Gender….) and a verified Google account.
1.2 To verify, the software will take a random verification code which will be generated and sent to the user's email address by the system.
1.3 After verification, the user must set their username and password.
1.4 The username must be unique, and the password should be strong.
1.5 If the username already exists in the database records, then repeat 1.3 with an alert message.
1.6 If the registration is successful, the login page of the user account will be displayed.

Priority Level: High
Precondition: The user has a valid google account
Cross-references:

2. Set goals

Functional Requirement

2.1 The site will take information about goals(task).
2.2 This site will take location regarding goals.
2.3 User can attach time if any offline possibility.
2.4 User can add collaborators (who want to work together) via email if wants.
2.5 Can **skip** a current task and reset it for later.
2.6 User can mark and display the completed task.

Priority Level: High
Precondition: Successful login
Cross-references: 3.3

3. Add socially

Functional Requirement

3.1 Will show all users of this application from the contact database and social sites database or contacts (WhatsApp, Facebook, Contacts etc.)
3.2 User can add friends externally by getting an invitation link.
3.3 Can see up-to-date rank by completed tasks.
3.4 User can share specific moments of their choice.

Priority Level: Medium
Precondition: Successful login and must be enabled

Cross-references: 2.4

4. Settings

Functional Requirement

4.1 The software shall allow users to enable default time for goal if there is any offline possibility at a certain location.
4.2 Can change the display options (e.g. add items from the bottom).
4.3 Can be able to enable or disable add socially.
4.4 Enable ghost mode to hide activities.
4.5 Edit profile (username, DP, password, etc.)

Priority Level: Low
Precondition: Successful login.
Cross-references: 2.3, 3

5. Searching

Functional Requirement

5.1 The system shall allow users to search for previously added goals/tasks.
5.2 The search feature shall enable the user to search based on the task name, location and collaborators.
5.3 The system shall suggest related tasks based on previous activity or missed activity.
5.4 The system shall use GPS to suggest tasks based on the user's current location and previous activity.
5.5 The system shall allow users to view missed tasks and suggest a reminder to complete them.
5.6 The system shall prioritize suggested tasks based on their proximity, urgency, and priority level.
5.7 The system shall allow users to filter search results based on completed or incomplete tasks.
5.8 The software shall allow users to sort search results based on date added and priority level
5.9 The system shall display the search results in a user-friendly manner with necessary details like task name, location, and collaborators.
5.10 The software shall allow users to modify previously added tasks based on the search results.

Priority Level: Medium
Precondition: Successful login and previously added tasks.
Cross-references: 2, 3

6. Snooze

Functional Requirement

6.1 The system shall allow users to snooze a reminder for a certain amount of time.
6.2 The snooze feature shall be available for incomplete tasks and upcoming reminders.
6.3 The system shall provide preset time intervals for snooze, such as 5 minutes, 10 minutes, 15 minutes, 30 minutes, and 1 hour.
6.4 The software shall allow users to customize the snooze time interval as per their preferences.
6.5 The system shall remind the user after the snooze period is over if the task is still incomplete.

Priority Level: Medium
Precondition: Successful login and reminders to be snoozed.
Cross-references: 2

7. Re-reminder

Functional Requirement

7.1 The system shall suggest missed tasks periodically and location based.
7.2 The periodic mode shall be customizable, based on the user's selected time interval, such as every hour or every day.
7.3 The location-based mode shall suggest missed tasks when the user is in the vicinity of the task location.
7.4 The system shall prioritize missed tasks based on their proximity, urgency, and priority level.
7.5 The system shall allow users to mark missed tasks as complete or snooze them for a later reminder.

Priority Level: High
Precondition: Successful login and missed tasks.
Cross-references: 2

3.2 Non-Functional/Quality Requirements:

QA1: Usability: *A trained user shall be able to set 5 goals(tasks) in a single visit of set-goals function in an average of two and a maximum of three minutes.*

- **Priority Level: High**
  Precondition: Minimum configuration android 5.0 and 32 bits
  Cross-references:

QA2: Interoperability: *The GPS-Based Reminder System shall be able to import any valid location information from google Maps or the MapQuest mapping tool.*

- **Priority Level: High**
  Precondition: email should well be verified
  Cross-references:

QA3: Integrity: *Only existing and verified emails will be granted to send the verification code to the user's phone.*

Priority Level: Medium
Precondition: The provided phone number should be active..
Cross-references

Important Primarily to Developers

QA4: Reusability: *The Add socially function shall be designed to be reusable at the object code level in other applications that may use for connecting people by existing connections.*

- **Priority Level: High**
  Precondition:
  Cross-references:

QA5: Testability: *The maximum cyclomatic complexity* of a module shall not exceed 20.*

- **Priority Level: Low**
  Precondition:
  Cross-references:

3.3 Project Requirements

Resources needed:

- Human resource(**people**) :
  - Numbers:
    Scrum master: 1 person
    Product Owner: 1 person
    Scrum team : 3-4 person (approx.)
  - Skill: More than 15 years' experience on s/w development field.
- Hardware/Networking devices(**environment**) :
  - **PC** : 6 (approx.)
  - **Network hub** : 1
  - **Network Switch**: 1
  - **Wireless access point**(router) : 1
  - **Ethernet Crossover cables** : 1 or 2
  - **Server** : 1
  - **Additional sources** (E.g.: Broadband connection)
- Reusable components(**reusable software**):
  - **Design documents**
  - **Modules**
  - **Function or object**
  - **Source code**
  - **User interfaces**
  - **DBMS**

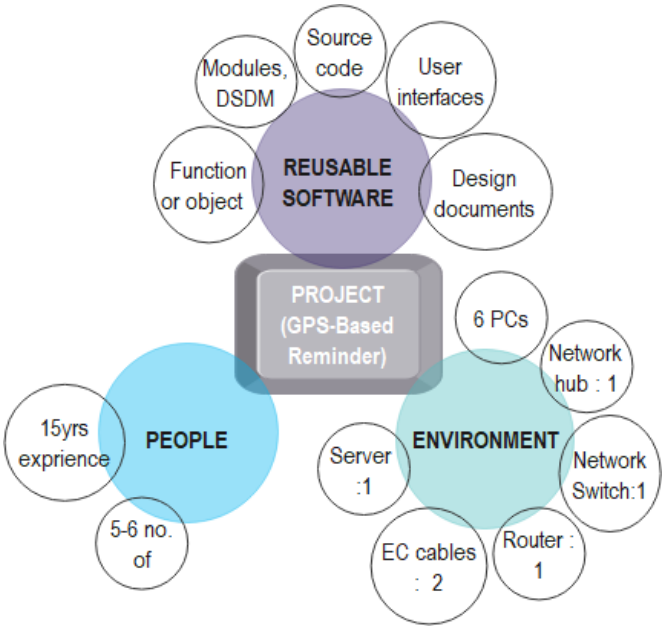


*Fig. **4**: Available resources and their allocation*

The micro-tasks related to project management:

Keys	Tasks	Keys	Tasks
A.	Define project goals and backlog list	L.	Design 'Seat Goal module
B.	Create a business case	M.	Develop 'Seat Goal module(f1,f2)
C.	Define sprints.	N.	Deliver prototype for 'Seat Goal module
D.	Create a project plan		…….
E.	Set a budget baseline	O.	Collect all modules and set multilevel relation
F.	Define roles and responsibilities.	P.	Apply integration test
G.	High-level design of the system	Q.	Check hardware integration
H.	Review meeting	R.	Check OS compatibility
I.	Design 'Registration' module	S.	Review scope document with customer
J.	Develop 'Registration' module(f1,f2)	T.	Check documents as required
K.	Deliver prototype for 'Seat Goal' module		

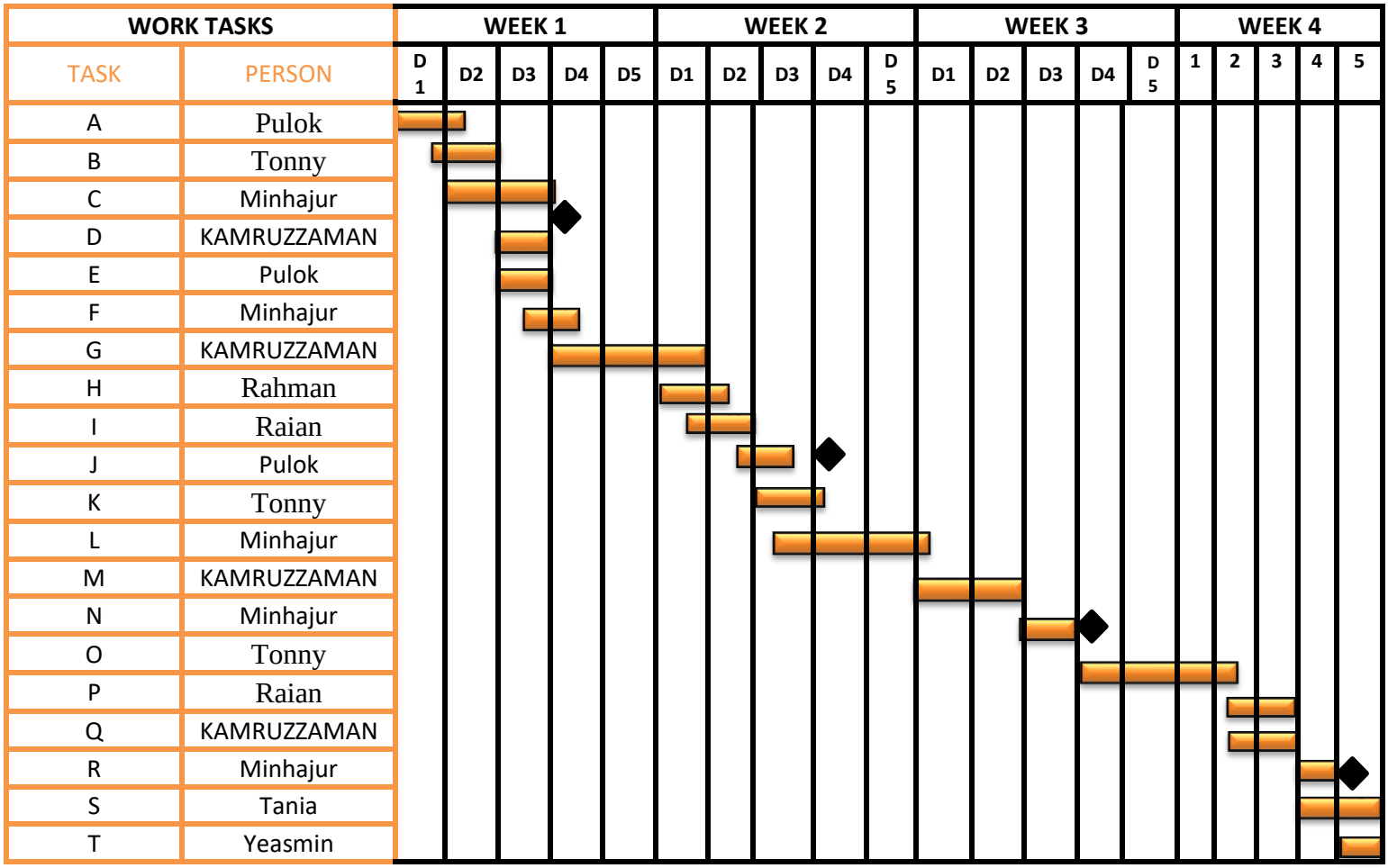


Fig. 5: Gantt chart for Project Scheduling

4. Design and Interface Requirements

4.1 UML Diagrams

Scenario for Use Case diagram: John needs to pick up his friend from the airport tomorrow at 5 pm. He opens the GPS-Based Reminder app and **login** into his account. Then, he uses the "**Set Goal**" function to locate the airport and set a reminder which will be triggered by GPS when he passes the airport. The app adds the reminder to John's list of **socially associated** tasks later it's also can "**search'**. On the day of the task, John is on his way to the airport when he receives a notification from the app reminding him to pick up his friend. On the way to the airport, John encounters unexpected traffic and realizes that he won't make it on time. He uses the "**Snooze**" function to delay the reminder until he reaches a point where he can reevaluate the traffic conditions and estimate his arrival time. Once John picks up his friend from the airport, he uses the "Archiving" function to remove the reminder from his list of tasks.

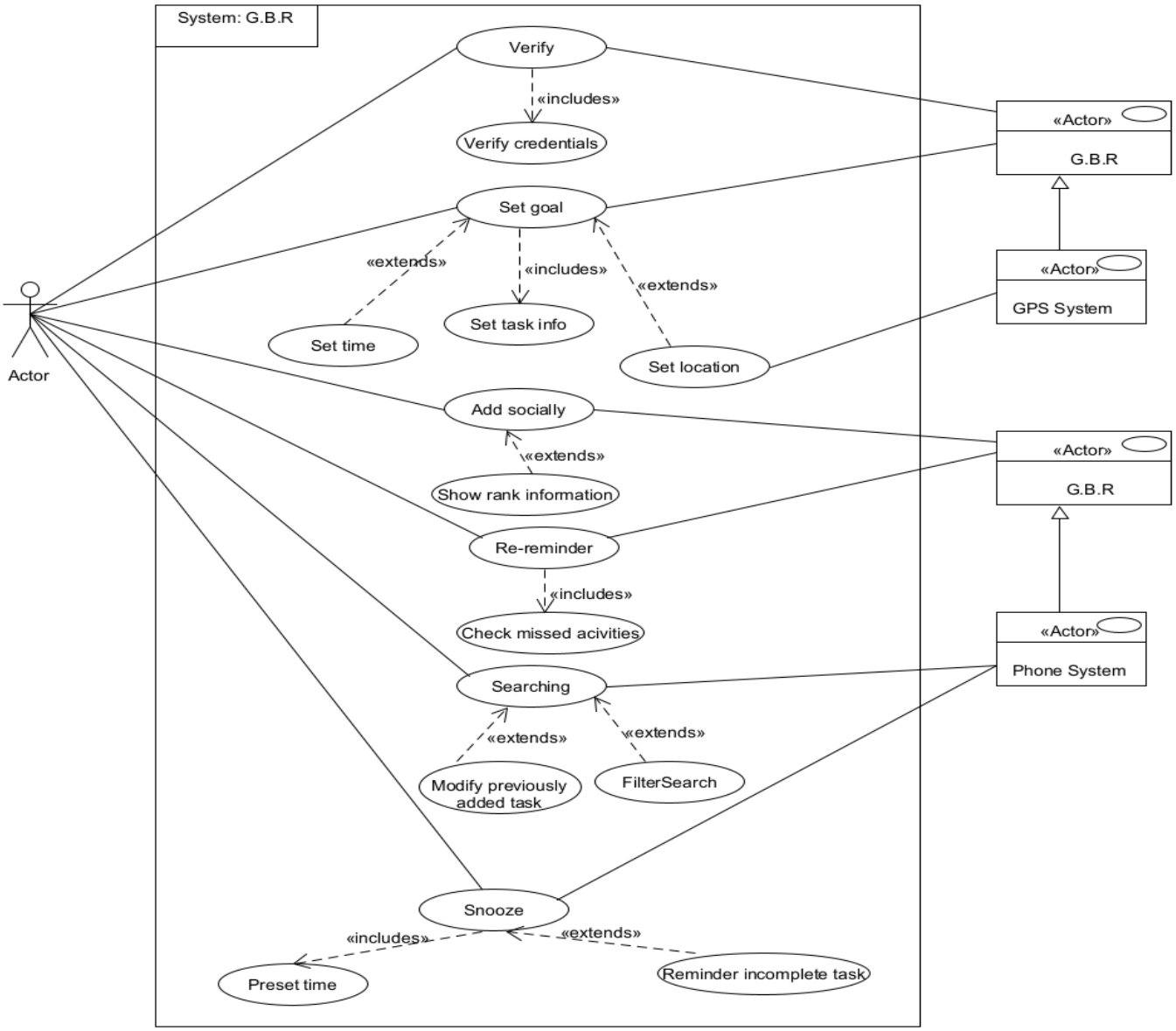


*Fig. **6**: Use case diagram*

Class diagram scenario for the user: User class will store information about the users of the system, such as username, password, and email. Goal class will store information about the user's goals, including the goal name, description, and deadline. Task class will store information about the user's tasks, including the task name, description, deadline, and status (completed or not). Reminder class will store information about the reminders set by the user for their goals and tasks, including the reminder time and frequency. Snooze class will provide a method to snooze reminders for a certain amount of time. Re-reminder class will provide a method to set periodic reminders for missed tasks. The settings class will store user-specific customization options for the system, such as display settings and notification preferences. The relationships between these classes can be represented as follows: The User class has a one-to-many relationship with the Goal and Task classes, as each user can have multiple goals and tasks. The Reminder class has a many-to-one relationship with both the Goal and Task classes, as each goal and task can have multiple reminders. The Search, Snooze, and Re-reminder classes are utility classes that provide additional functionality to the system and do not have any direct relationships with other classes. The Settings class has a one-to-one relationship with the User class, as each user can have only one set of customization options. Overall, this class diagram provides a clear representation of the system's core functionalities and their relationships, allowing for efficient and effective software development.

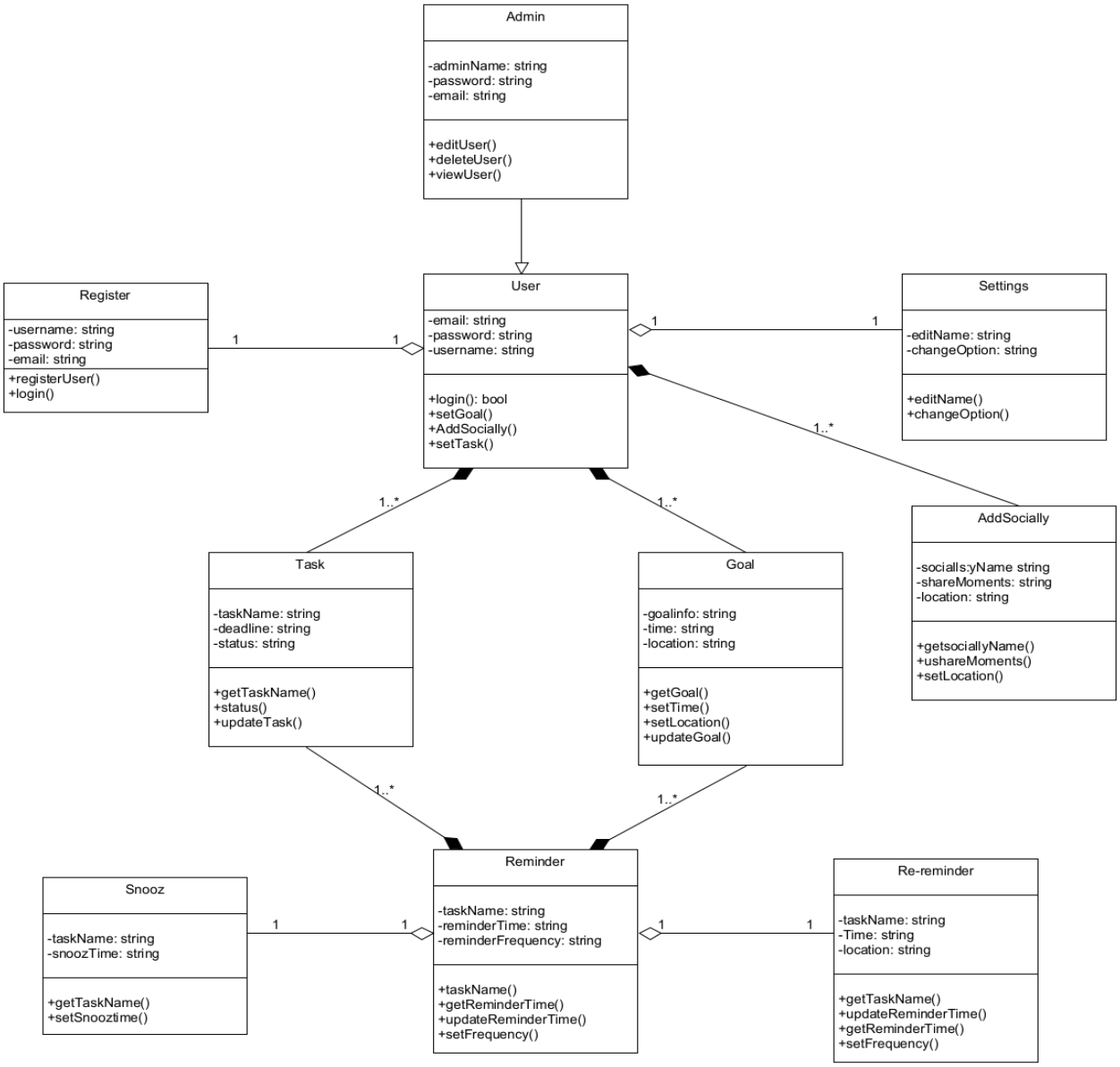


Fig. 7: Class diagram

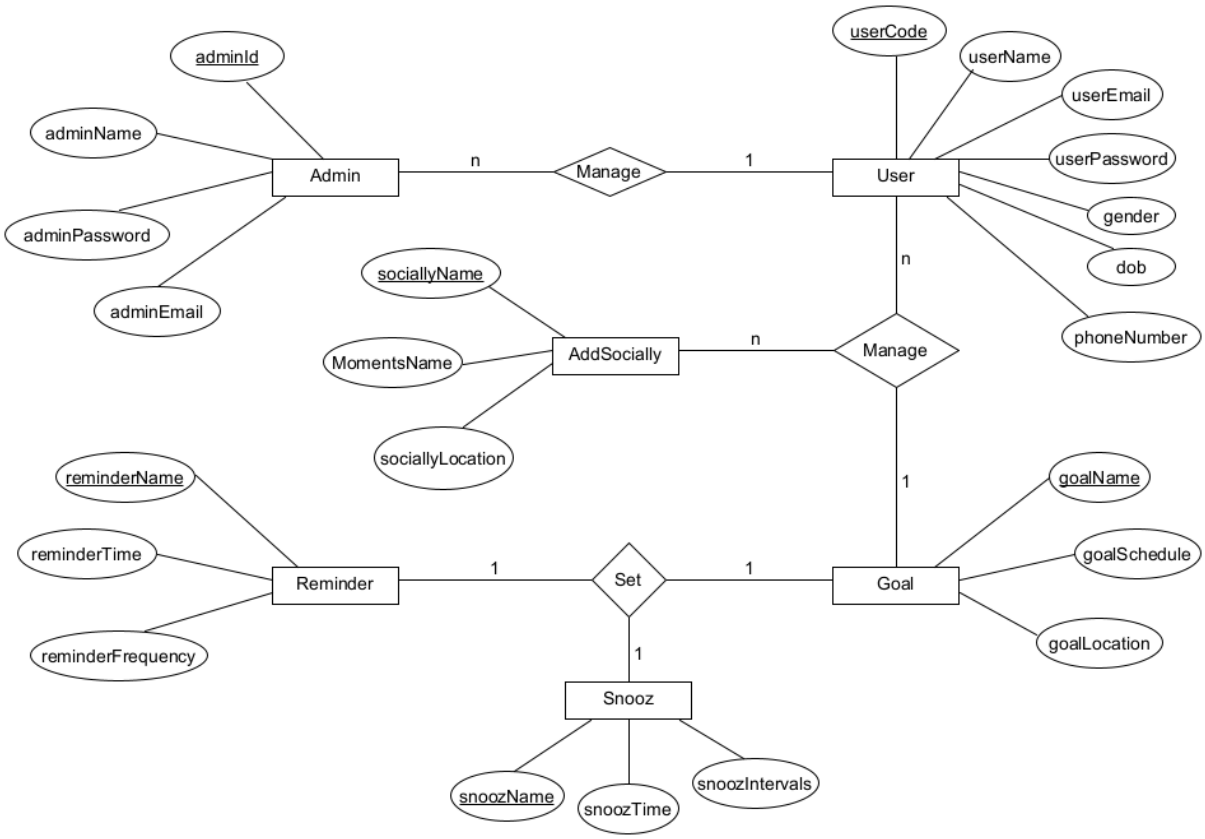


*Fig. **8**: ER diagram*

<u>Activity diagram scenario for the user:</u> The user logs in to the system. Once logged in, the user is redirected to the dashboard where they can see all their previously set goals. If the user wants to add a new goal, they click on the "Create" button. The user enters the goal name, description, and the deadline for the goal. The system verifies the input and saves the new goal to the user's account. The user can now see the newly added goal on their dashboard. If the user wants to update an existing goal, they click on the goal and select the "Update" option. The user can make the necessary changes to the goal and save it. The system verifies the input and updates the goal on the user's account. If the user wants to delete an existing goal, they click on the goal and select the "Delete" option. The system prompts the user to confirm the deletion. If the user confirms the deletion, the system removes the goal from the user's account. If the user wants to snooze a reminder for a particular goal, they can click on the goal and select the "Snooze" option. The user can then choose the snooze duration and save it. The system updates the reminder for the goal with the new snooze duration. If the user misses a task, the system sends a re-reminder to the user after a certain period, reminding them of the missed task.

This activity diagram scenario covers the main functionalities of the system, including setting and managing goals, setting, snoozing reminders, and receiving re-reminders for missed tasks.

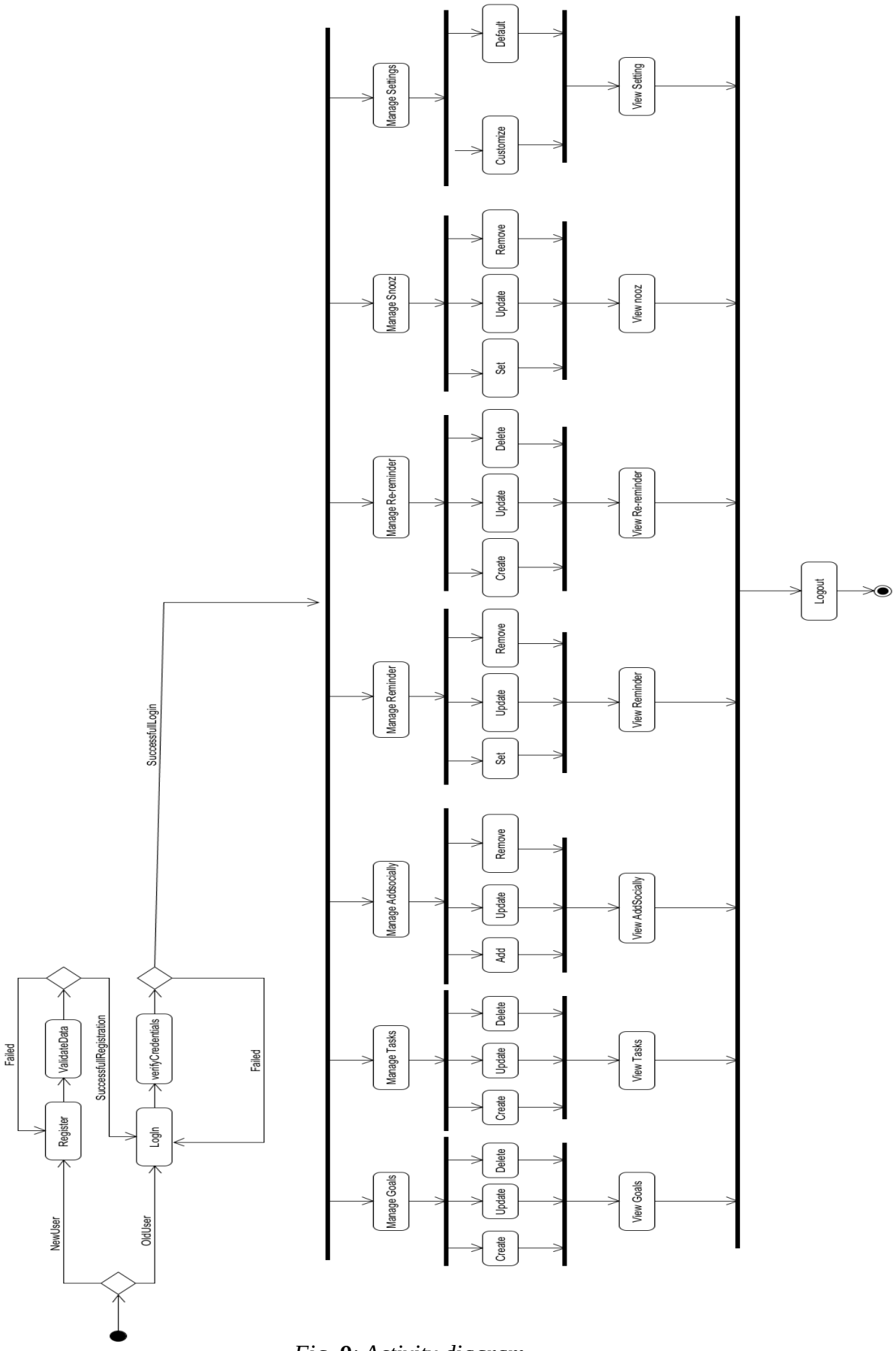


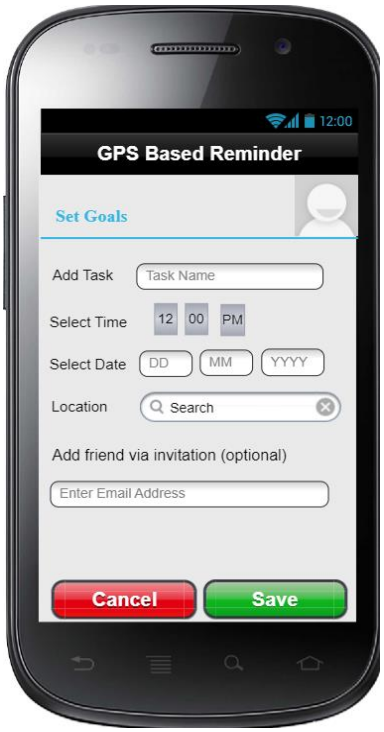
*Fig. **9**: Activity diagram*

4.2 Data Dictionary

Entity	Attribute	Type/Size	Validation	Key
User	userCode	Number(8)	10000-99999	Primary
User	Name	Text(30)	Required	
User	UserName	Text(10)	Required	
User	Password	Text(20)	Required	
User	DOB	Date(8)	Valid Date	
User	Gender	Text(6)		
User	PhoneNo	Number(15)		
User	Email	Text(35)	Required	
Admin	adminId	Number(8)	10000-99999	Primary
Admin	adminName	Text(30)	Required	
Admin	adminPassword	Text(20)	Required	
Admin	adminEmail	Text(35)	Required	
Goal	goalName	Text(30)	Required	Primary
Goal	goalSchedule	Text(35)	Required	
Goal	goalLocation	Text(20)	Required	
Reminder	reminderName	Text(35)	Required	Primary
Reminder	reminderTime	Text(30)		
Reminder	reminderFrequency	Number(8)	Required	
Snooz	snoozName	Text(35)	Required	Primary
Snooz	snoozTime	Text(20)	Required	
Snooz	snoozIntervals	Number(8)	Required	
AddSocially	SociallyName	Text(35)	Required	Primary
AddSocially	MomentsName	Text(30)		
AddSocially	sociallyLocation	Text(35)		

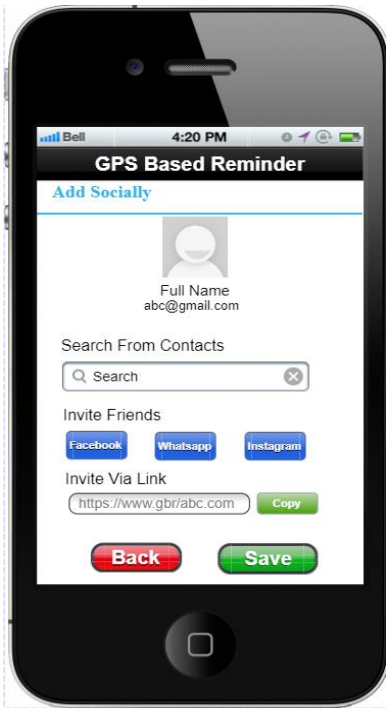
4.3 UI/UX Design Specification

Registration



Set Goal

Settings



Add Socially

*Fig. **10**: UI/UX Design by pencil tool*