Software Design Documentation

Ineed App: “Find what you are need”

Prepared By:

Rabia KIRIM
Büşra GÜL

january 3rd, 2017
Yıldırım Beyzıt University - Ankara

1.Scope

- This report will define the high level design and technology decisions of Ineed App.
- This sdd defines and describes the use of each view, the architectural constraints of the system, the functional requirements with a significant impact on the architecture, use-case realization, concurrency aspects, the layers and subsystems of the application, performance issues and constraints.

2. References

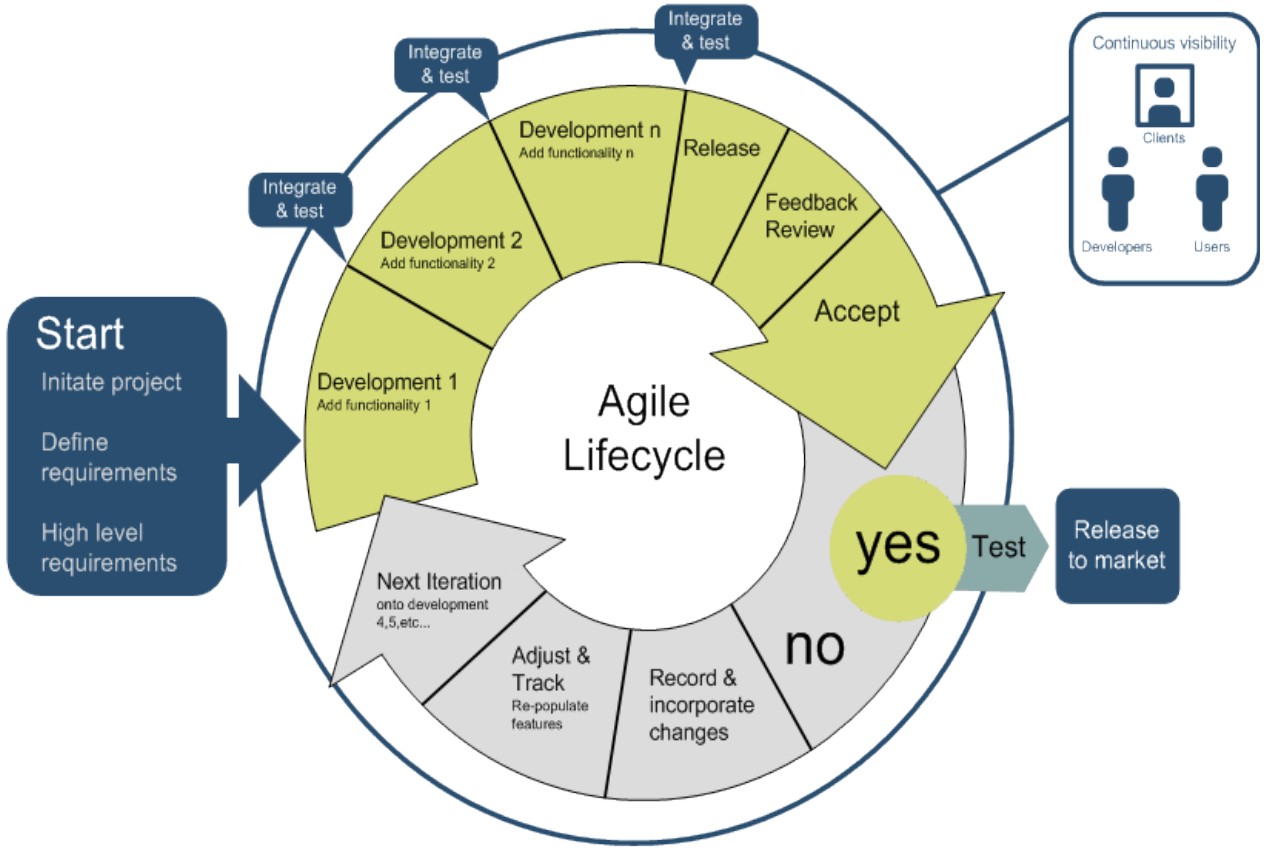
[14] Software Engineering Standards Committee of the IEEE Computer Society, "IEEE Recommended Practice for Software Design Descriptions", IEEE Std 1016-1998.

3.Definitions

In campus life people sometimes need some things like pencil, ruler, calculator etc. Instead of going and asking for people to find needed item, a mobile application will be used for this purpose. When a person need something they will specify the need in the mobile app and notifications will be sent to nearby people. People can chat within the app to meet and resolve the need. They can also see the others' position (optional) corresponding to their location. (i.e 2 minutes to your position) Helpers will get points.

4. Considerations for producing an SDD

4.1 Software life cycle



4.2 SDD within the life cycle

In the design part of our agile life cycle we define our sdd. gile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements. In agile the tasks are divided to time boxes (small time frames) to deliver specific features for a release.Iterative approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features; the final build holds all the features required by the customer.

SDD:

The purpose of this sdd is to define and describe the use of each view, the architectural constraints of the system, the functional requirements with a significant impact on the architecture, use-case realization, concurrency aspects, the layers and subsystems of the application, performance issues and constraints.

5. Design Description Information

Introduction

This document is adopted from the Software Engineering Standards Committee of the IEEE Computer Society, *"IEEE Recommended Practice for Software Design Descriptions"*, IEEE Std 1016-1998.The Software Design Descriptions (SDD) provides an architectural overview of Online Carpooling System. This document presents to various stakeholders different types of abstraction It aims to provide the stakeholders a clear understanding of the system.

Scope

The scope of this SDD is to define high level design and technology decisions of the Ineed App.

Architectural Representation

The views used to document the Ineed App in current phase are:

Use Case view

User: all the stakeholders of the system, including the end-users.
Area: describes the set of scenarios and/or use cases that represent some significant, central functionality of the system.
Related Artifacts : Use-Case Model, Use-Case documents

Architectural Goals and Constraints

This section describes the software requirements and objectives that have some significant impact on the architecture.

Technical Platform

Ineed application will be deployed onto a Android Studıo application server.

Transaction

Android Studıo application platform already has built in transaction capabilities, they will be used.

Security

The system must be secured, so that a customer can make online payments (Premium Membership)

Basic security behaviors:

- Authentication: Login using at least a user name and a password
- Authorization: according to their profile, online user must be granted or not allowed to receive some specific services (Automatic match finding, Ride Suggestion, etc...)

Persistence

Data persistence will be addressed using a relational database and Android Studıo applications Object Relational Mapping capability will be reused.

Reliability/Availability

High availability is required since there are monetary issues related to the systems availability. Advertisers and premium users should not be disappointed. The system's high availability will also ensure customer satisfaction and loyalty. Android Studıo application solutions will be used.

Targeted availability: 16 hours a day, 7 days a week (Maintenance at night)

Performance

Search queries should return %90 of the time below 5 sec.

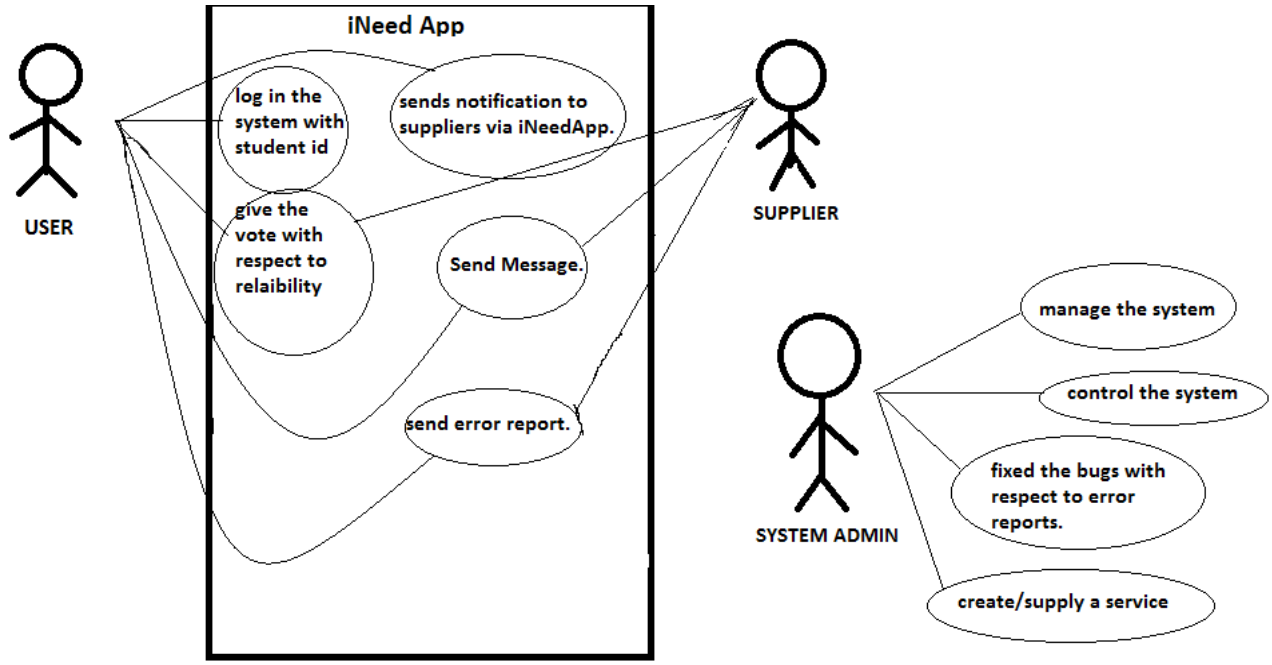
Credit card payment transaction should finish in 10 sec.

Internationalization

Initially the system will support Turkish.

Use-Case View

This section provides a functional overview of the system by a use-case diagram.



Use-Case Realizations

Not the scope of this phase.

Logical View

Details

- Development is done on an extra application server
- Database server is SQlite server.
- Use Android Studio platform for development.

Quality

Scalability:

- **Description** : System's reaction when user demands increase
- **Solution** : System will built to support too many members, further scalability is supported by new hardware

Reliability, **Availability**:

- **Description** : Transparent failover mechanism, mean-time-between-failure
- **Solution :** : Ineed application server supports load balancing.

Portability:

- **Description** : Ability to be reused in another environment
- **Solution :** The system is fully application compliant and thus can be deployed onto any application server

Security:

- **Description** : Authentication and authorization mechanisms
- **Solution :** application native security mechanisms will be reused

Risks and Mitigation Plans

Software failure

- Development/Operational/Database server fails

  Recover using backup

- Development workstations fail.

  Use spare workstations if not fixed in short time.

6. Design description organization

6.1 Introduction

The Software Design Description (SDD) details the chosen software architecture and the justification for selecting that architecture. In this project the team was tasked with architecting and implementing a Ineed application System.

6.2 Design views

Entity attribute information can be organized in several ways to reveal all of the essential aspects of a design. In so doing, the user is able to focus on design details

from a different perspective or viewpoint. A *design view* is a subset of design entity attribute information that is specifically suited to the needs of a software project activity. Each design view represents a separate concern about a software system. Together, these views provide a comprehensive description of the design in a concise and usable form that simplifies information access and assimilation. A recommended organization of the SDD into separate design views to facilitate information access and assimilation is given in Table 1. Each of these views, their use, and representation are discussed in detail.

6.2.1 Decomposition description

6.2.1.1 Scope

We first must understand our organization's business functions before beginning developing systems. The decomposition descriptions are done in order to plan business functions, processes, and sub processes within “INEED”Project.

6.2.1.2 Use

Our Project will perform a variety of different functions. Before we plan what systems to build for the organization, it is helpful to first understand the functions of “INEED” Projects needs to perform. Then it is much easier to identify processes that occur within the business functions, and ultimately the systems that will support those processes. This is a top-down approach to systems development. (As we mentioned we are going to use a Top- Down Agile like design in earlier stages.)

The process of starting at a high level and moving into smaller and smaller subsystems is called decomposition. The functional decomposition diagram (FDD) is a planning tool for identifying business functions and the processes that comprise them. The diagram is the starting point for more detailed process diagrams, such as data flow diagrams

Objectives

- Understand the rules and style guidelines for functional decomposition diagrams (FDDs).
- Understand the process used to create FDDs.
  2. Be able to create a Functional Decomposition Diagram.

6.2.1.3 Representation

USER

USERID	NAME	SURNAME	PASSWORD	EMAIL

HELP

HELPID	FROM	TO	DATE	ITEM

VOTE

USERID	HELPID	RATE	COMMENT

ADMIN

USERID

6.2.2 Dependency description

6.2.2.1 Scope

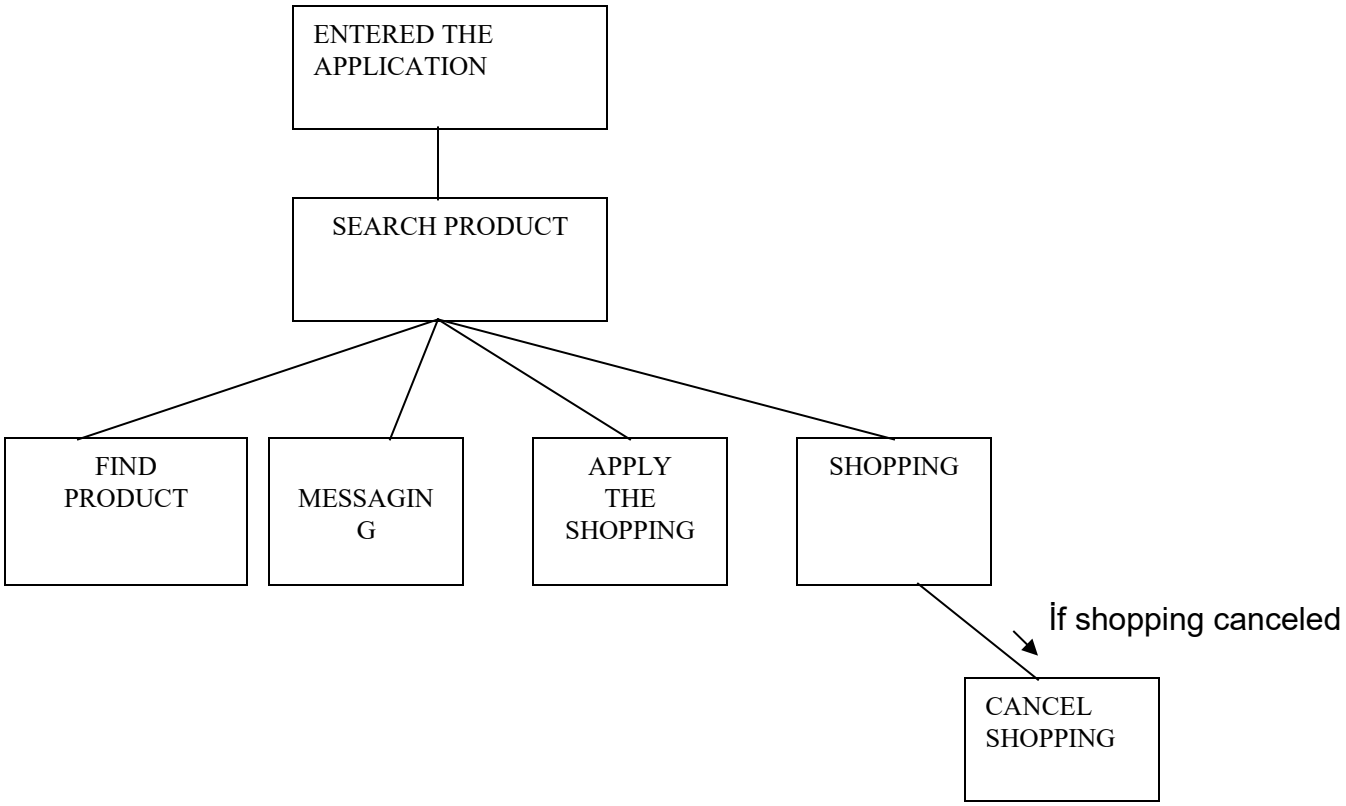
In our Project many methods are available for visualization and measurement of structure during the design of a programming solution.

6.2.2.2 Use

In order to avoid the possible problems might occur due to visualizing the program structure we draw as many diagrams as possible since the diagrams are more clear and easily understandable under any case

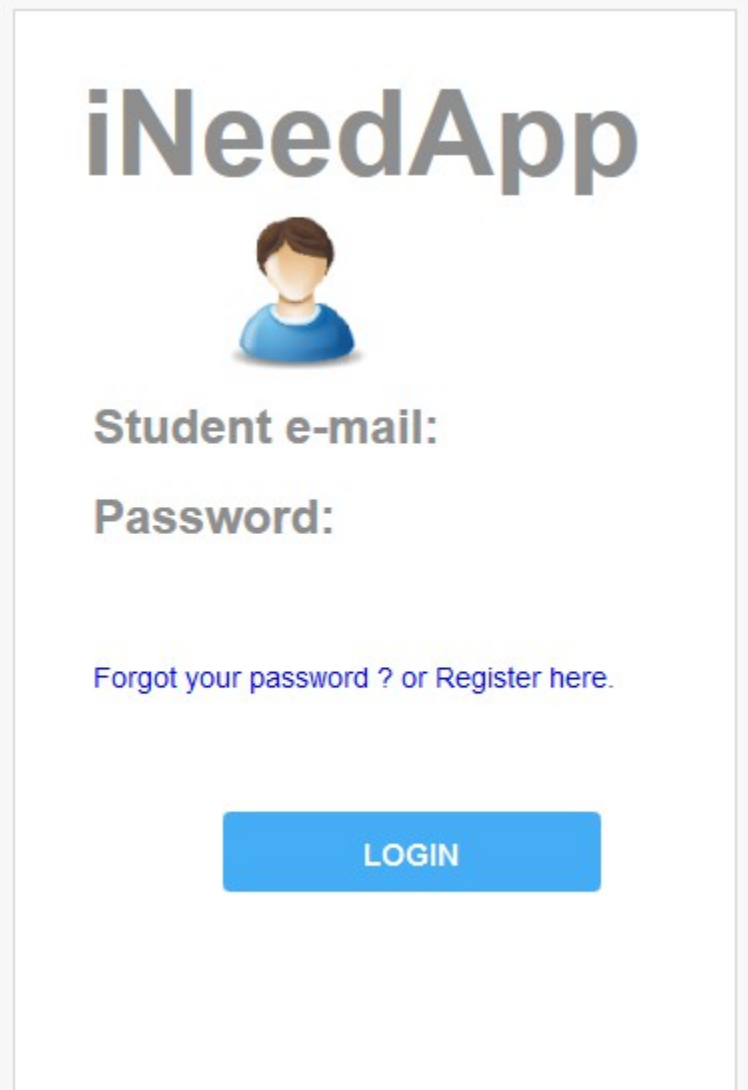
6.2.2.3 Representation

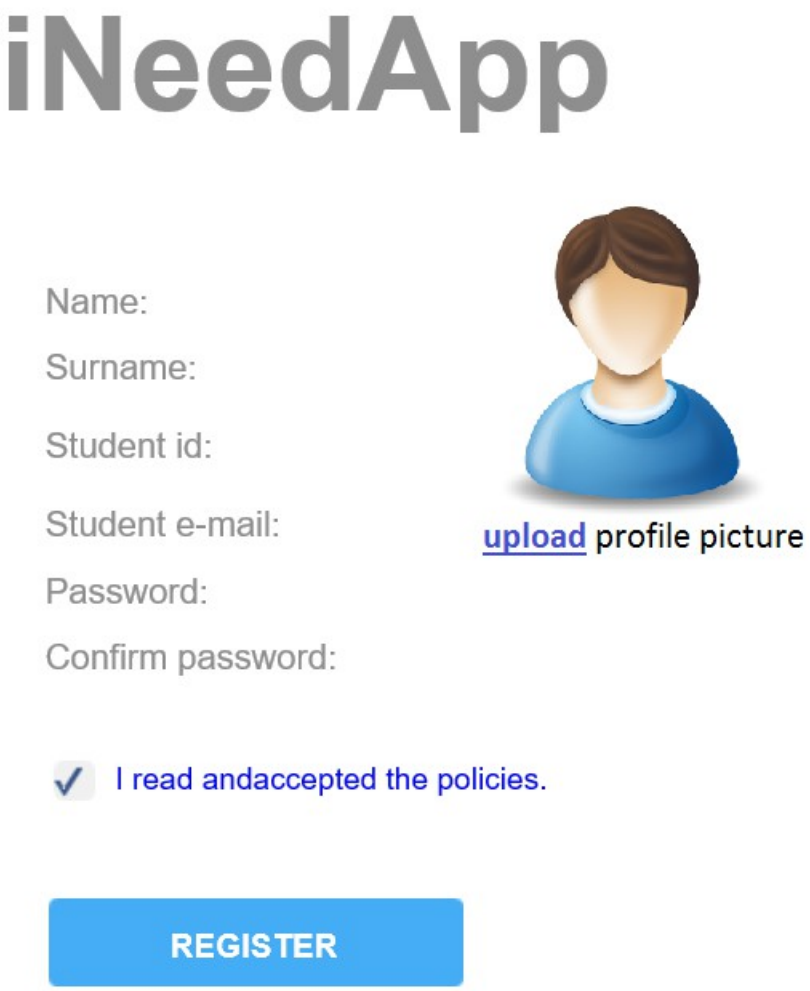
STRUCUTRE CHART

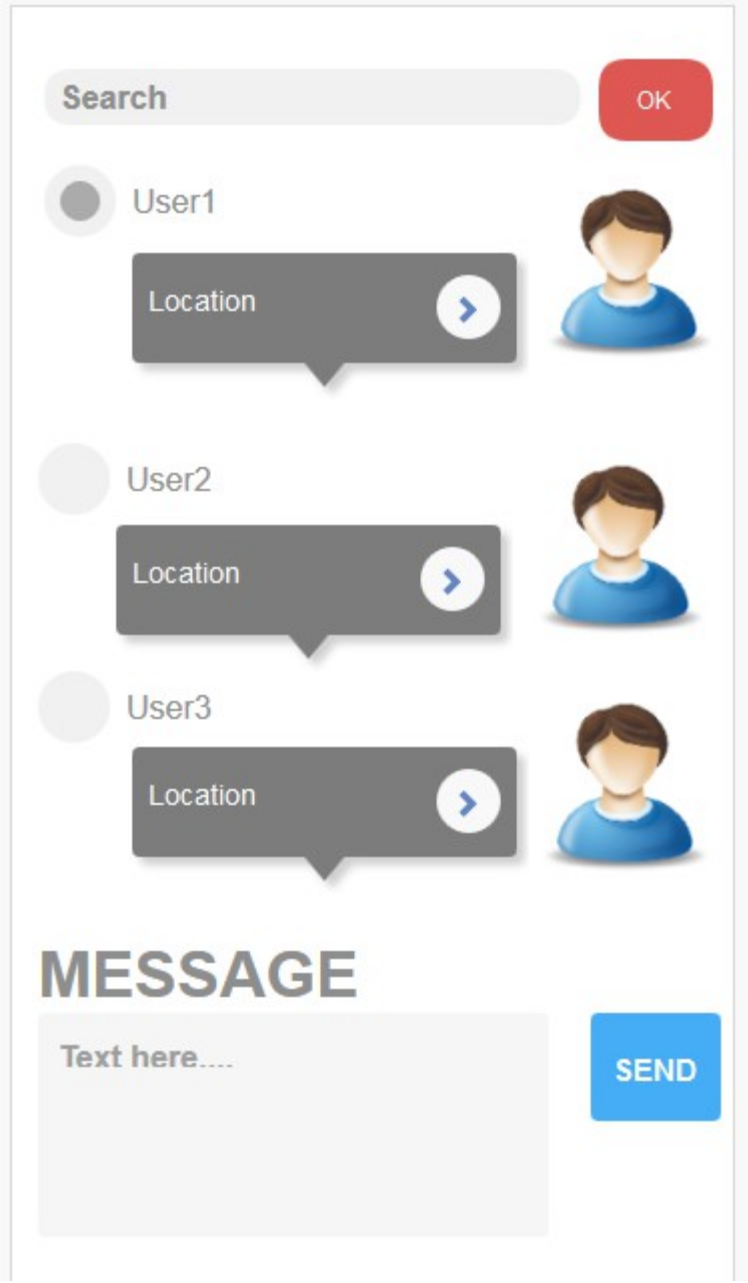


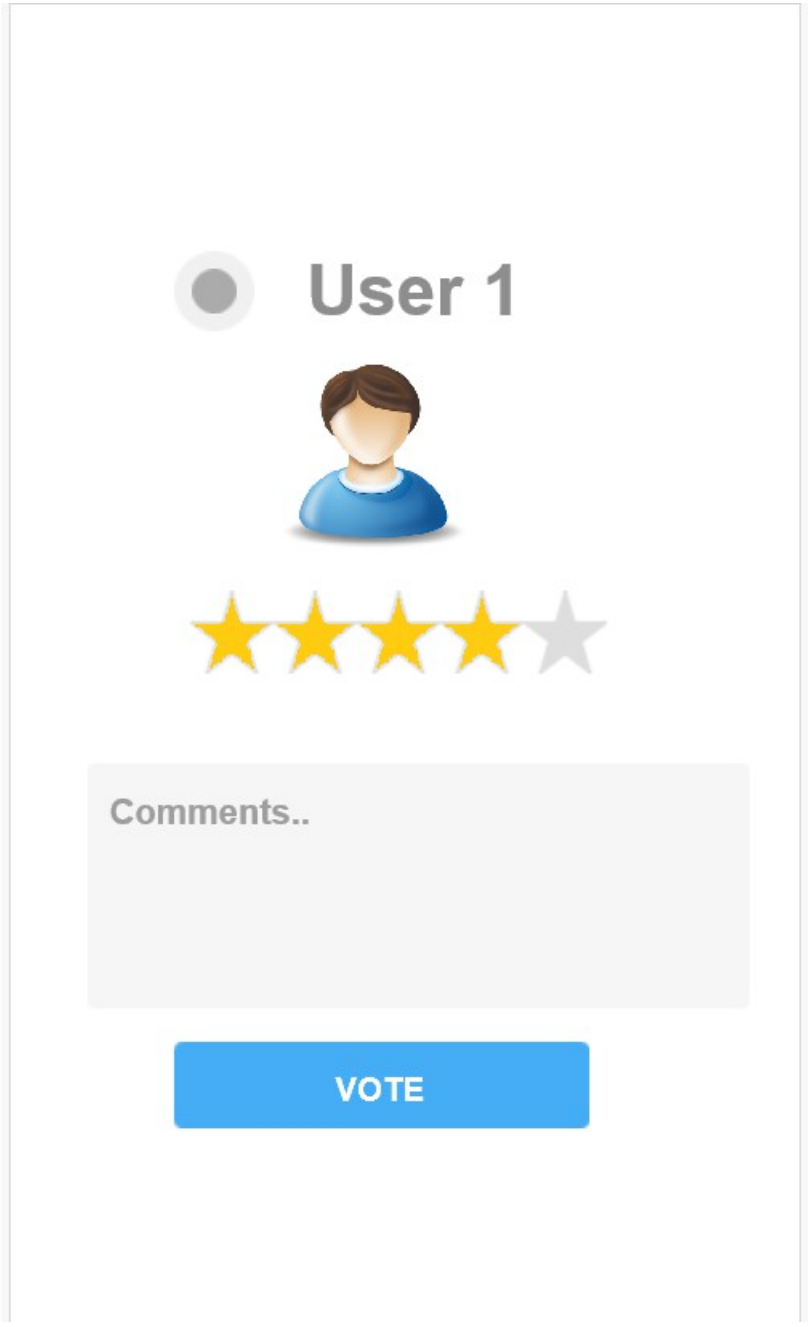
6.2.3 Interface description

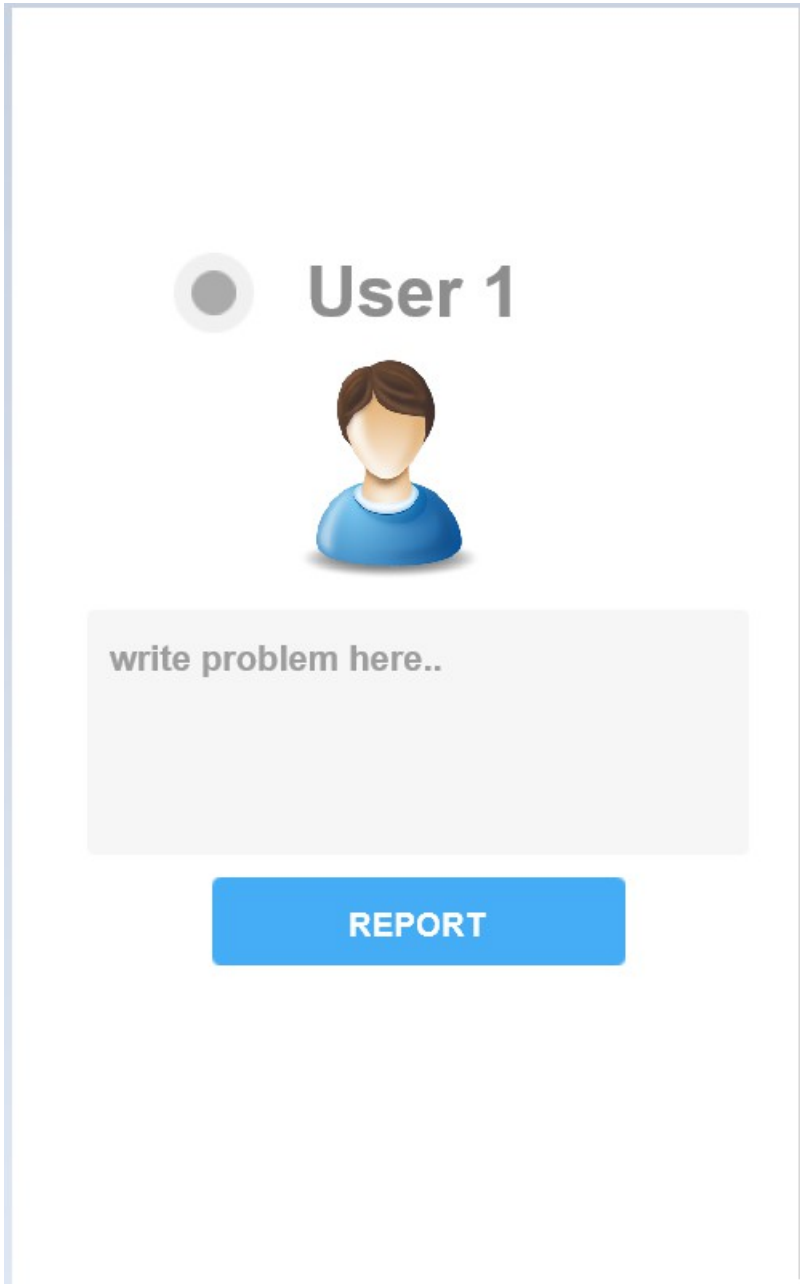
6.2.3.1 Representation




iNeedApp
Name:
Surname:
Student id:
Student e-mail:
Password:
Confirm password:
upload profile picture
I read andaccepted the policies.
REGISTER


Search
OK
User1
Location
User2
Location
User3
Location
MESSAGE
Text here....
SEND


User 1
Comments..
VOTE


User 1
write problem here..
REPORT