Bilkent University

Object-Oriented Software Engineering

Department of Computer Engineering

Roomie

Analysis Report

FIRST DRAFT

Project Group:

- *Deniz Alkışlar*
- *Ekinsu Bozdağ*
- *Eliz Tekcan*
- *Serhat Aras*
- *Selen Haysal*

Supervisor:

- *Bora Güngören*

Group 3.K| CS 319/2| 7 October 2017

Contents

1. Introduction 2

1.1 Purpose 2

1.2 Scope of Project 2

1.3 Glossary 3

1.4 References 3

1.5 Overview of the Document 3

2. Overall Description 3

2.1 Product Perspective 3

2.1.1 System Interfaces 3

2.1.2 User Interfaces 4

2.1.3 Hardware Interfaces 13

2.1.4 Software Interfaces 13

2.1.5 Memory Constraints 13

2.1.6 Operations 13

2.1.7 Site Adaptation Requirements 13

2.2 Product Functions 14

2.3 Use Case Scenarios 16

2.3.1 Story Scenarios 16

2.3.2 Active Event Scenarios 18

2.3.3 Die Scenarios 18

2.3.4 Win Scenarios 18

2.3.5 Extreme Scenarios 19

2.3.5 Expense Scenarios 19

2.3.6 Randomized Item Drop Scenarios 19

2.3.7 Status Boost Scenarios 19

2.3.8 Status Bar Changing “Prize” Scenarios 19

2.4 User Characteristics 20

2.5 Constraints 20

2.6 Assumptions and Dependencies 20

2.7 Apportioning the Requirements 20

3. Specific Requirements 21

3.1 External Interface Requirements 21

3.1.1 User Interfaces 21

3.1.2 Hardware Interfaces 21

3.2 Functional Requirements 21

3.3 Performance Requirements 22

3.4 Design Constraints 23
3.5 Software Design Attributes.. 23
3.5.1 Reliability.. 23
3.5.2 Availability 23
3.5.3 Maintainability 23
3.5.4 Portability 23
3.5.6 Extendibility 24
3.5.7 User-friendliness 24
4. Appendixes 24
5. References... 27

1. Introduction

1.1 Purpose

The intention of this document is to present an overall and comprehensive description of our game - Roomie. It will demonstrate the features and scope of Roomie, along with information about the requirements and constraints. The user cases and their interactions with the game will also be revealed in detail. This analysis report is an essential milestone in our project since throughout the process, it will serve as a guideline for the developers and provide extensive information for the clients.

1.2 Scope of Project

Roomie is a single player, story-based Android game that is designed to simulate the life of a university student. The game proceeds as the player makes choices while answering the prompted questions- each possible answer leads the player to another scenario. At the very beginning of the game, the player is expected to create a character by choosing a username and a gender. Roomie offers its users an extraordinary story line and various unexpected endings. Each player is equipped with a ‘backpack’ where he can use/sell the items that he has collected throughout the game. A status bar is devised to keep the users engaged and competitive in the game, it displays four different labels of statuses: health, money, sociality and grades. The choices that the user makes or the circumstances that the user faces may decrease or increase the levels of these statuses.

1.3 Glossary

Throughout the document, the term “backpack” means the inventory of the user.

1.4 References

Salen, Katie, and Eric Zimmerman. Rules of Play: Game Design Fundamentals. The MIT Press, 2010.

1.5 Overview of the Document

In the Overall Description section of this Analysis Report, the main functionalities of the game are explained, along with some extensive information about user-interfaces, user cases, and some basic details about requirements. In the Specific Requirements section, the details about requirements and constraints will be extended. Please see the appendices for our UML and Sequence diagrams.

2. Overall Description

2.1 Product Perspective

Roomie is an Android application which is a story-based game and it requires user interaction to proceed through the story-line. It does not require external access to any servers or databases, and it can be played offline. The user and her/his interaction is essential since the game is basically a real-life simulator.

2.1.1 System Interfaces

As an Android application, the only system interfaces Roomie provides are the touch-screen of the players’ phones and some of the phone’s own action listener sensors. The interactions that are done by the means of the touch-screen will be used in Roomie, meaning that the user interface will be displayed on the touch screen. The users can use their phones’ built-in buttons to exit from the game, or change the volume of the sound. Besides, while playing the game, the sensors that are built in their phones will be used to understand if the users are engaging in completing the missions that are assigned. For instance, the players are required to keep silent for at least ten second while they are in the “Library” mission. Roomie will use Android’s sensors to implement this kind of missions. Also the phone's' speakers will be used to play the game's sounds.

2.1.2 User Interfaces

Roomie consists of two main activities. One of the activities is the welcome activity. Welcome activity is shown on the first launch of Roomie. Welcome activity has a text input which asks name of the user. It also has radio buttons to take gender of the user.

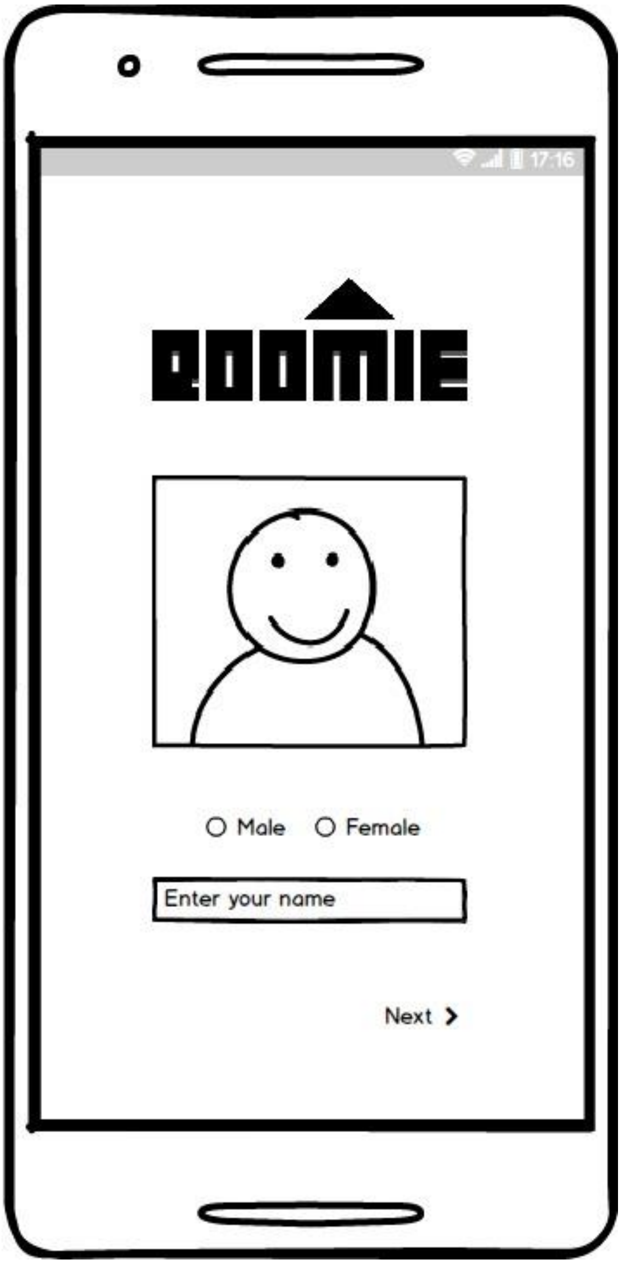


Figure: A Mock-up of the scene where the user creates his character(Welcome Activity)

The other activity is main activity. Main activity is where Roomie proceeds. The screen has three parts which are top, middle and bottom. On the left top there is a picture of the player. It is also a button which shows the current question as a dialog when it is clicked. On the right of the player’s picture there is a button for showing items in the player’s backpack. On the right top, there is a button to mute sound of Roomie. Middle part of the screen shows rooms of the player’s house. It shows one room at a time and the user could switch between rooms by using buttons at below. Finally, on the bottom part there are four indicators which show player’s current status.

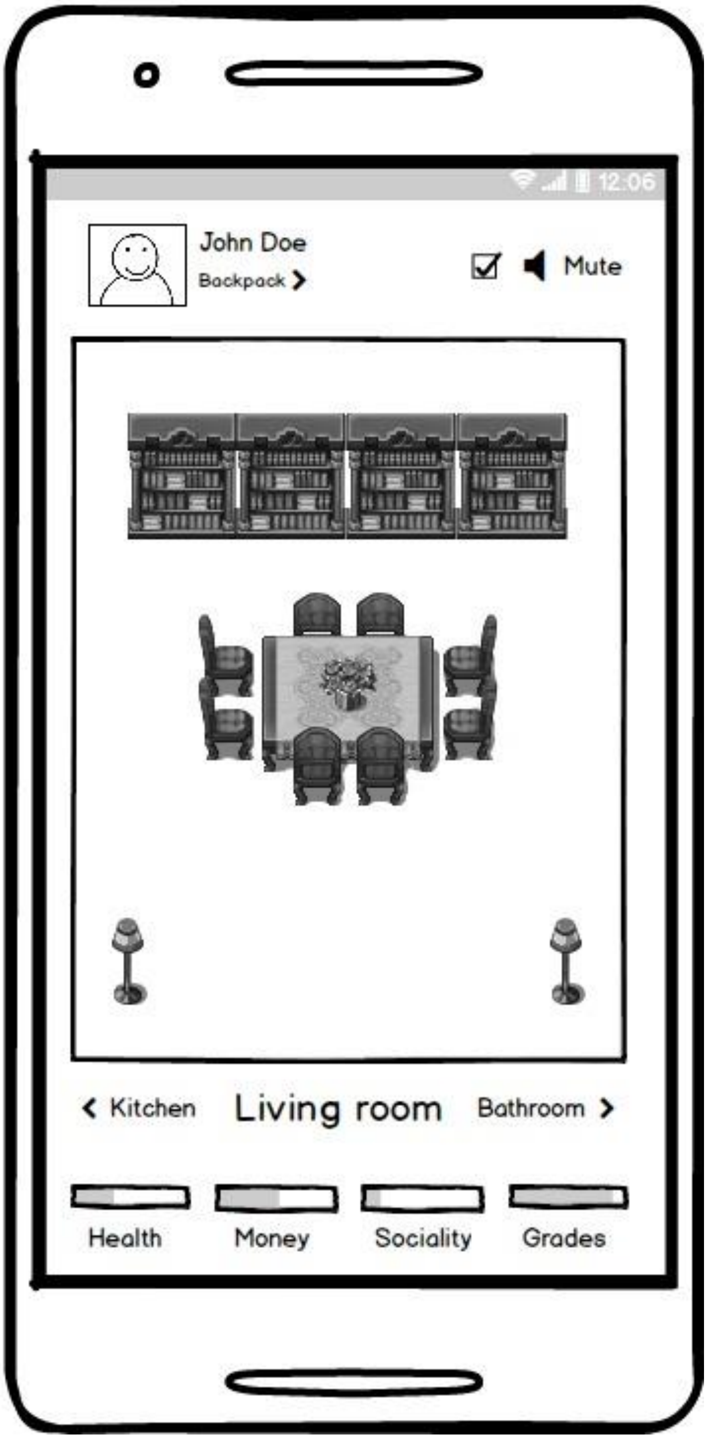
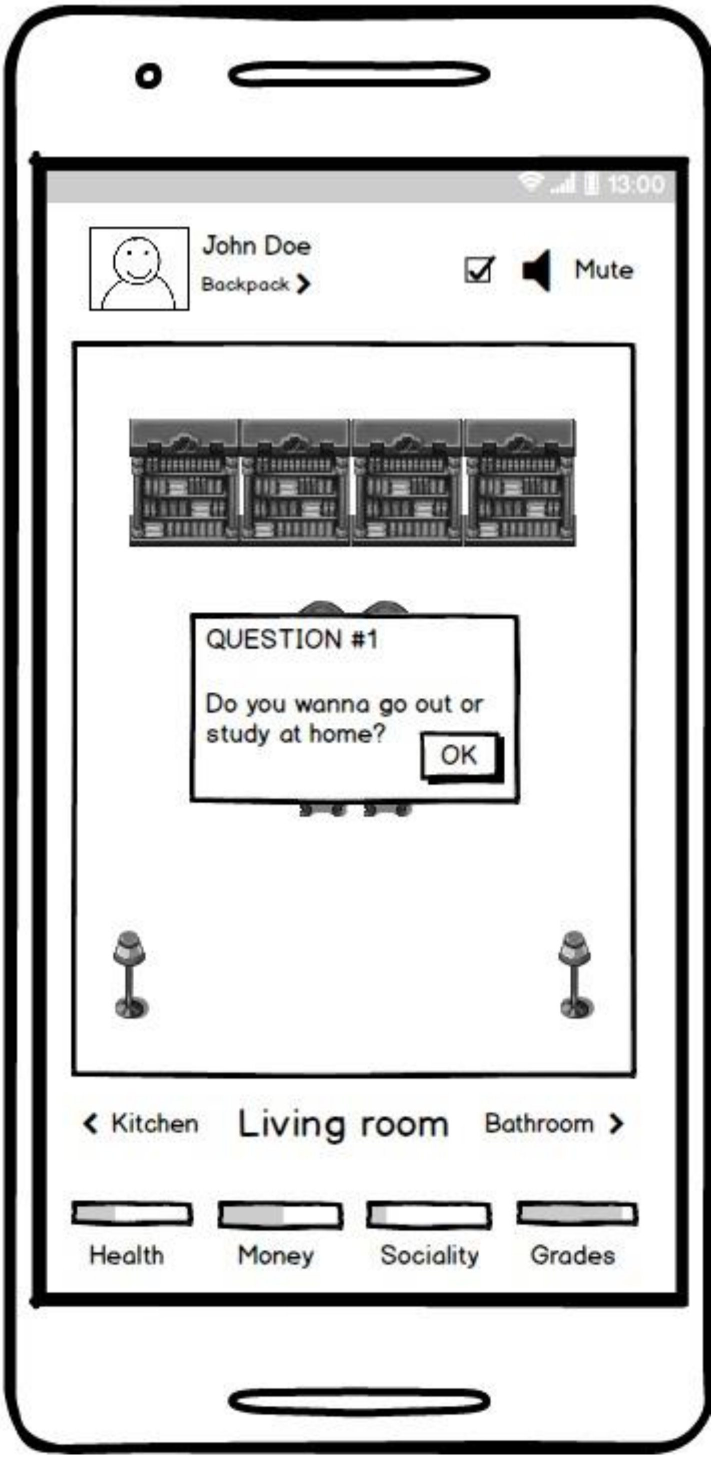


Figure: A Mock-up screening the main game scene, the Student Flat



QUESTION #1

Do you wanna go out or study at home?

Figure: A Mock-up screening a question that is prompted

The questions are shown as dialog box. The user is going to click items in the rooms to answer the questions. For instance, if one question is “do you want to go out or study at home?” the user is going to click door for going out and click desk for studying.

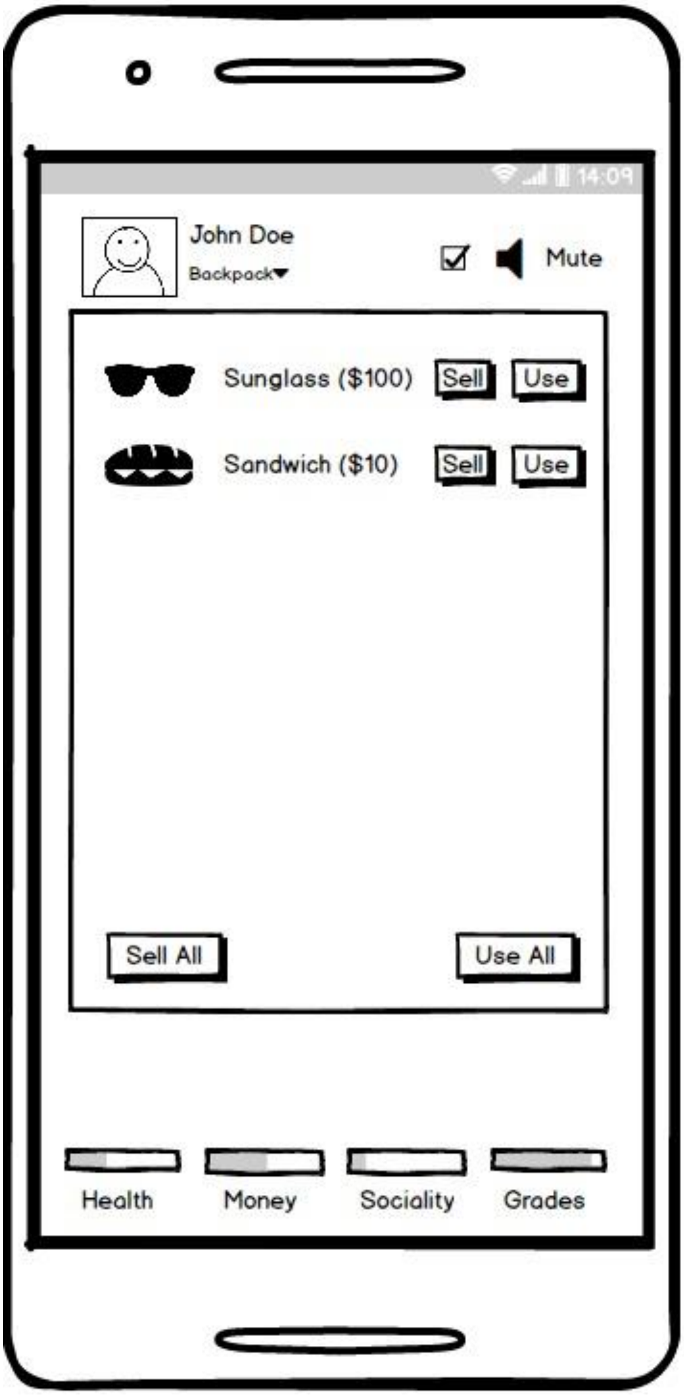


Figure: A Mock-up screening the 'backpack' of the player

Backpack of the player is shown when the user clicks the button on the top. Items in the backpack will be shown as a list. The user could use or sell a specific item from the list. If the player uses an item, indicators will be changed. If the player sells an item, money indicator will be increased. Using or selling all the items at once is also possible with the buttons at below.

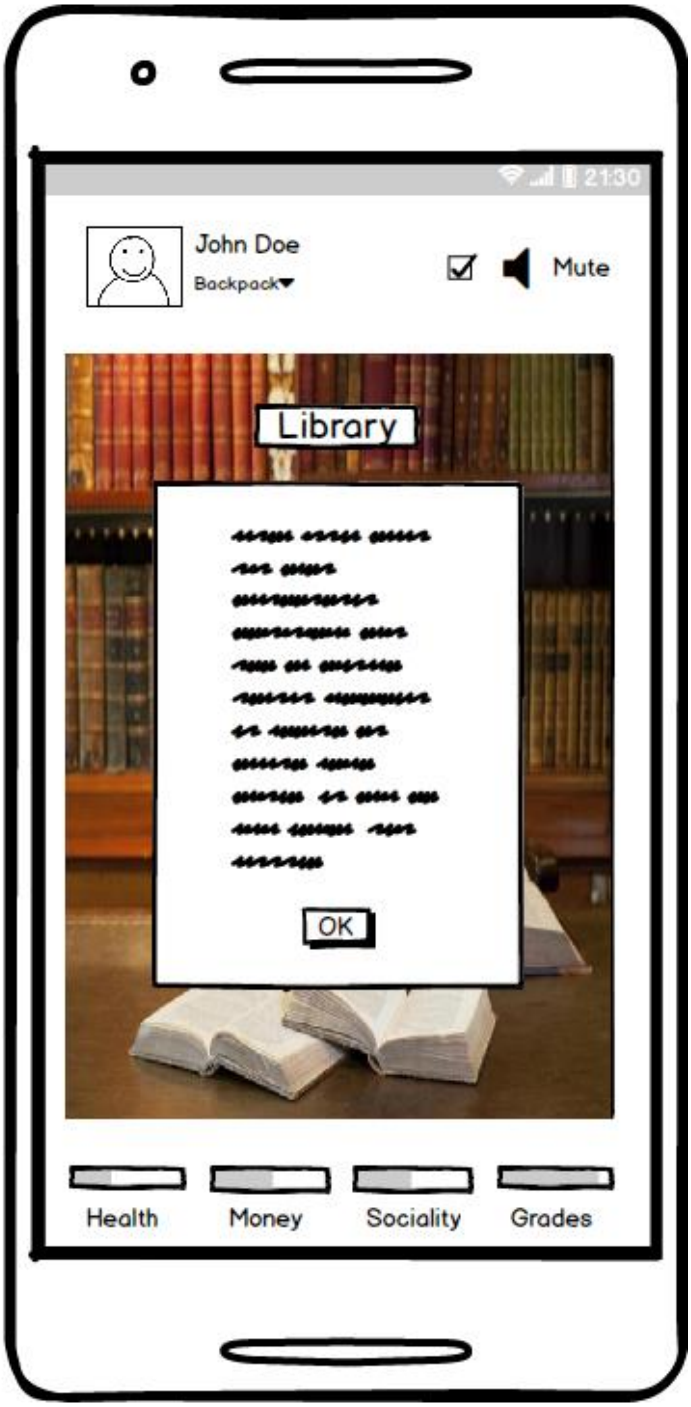


Figure: A Mock-up screening the “Library” scene of the Roomie

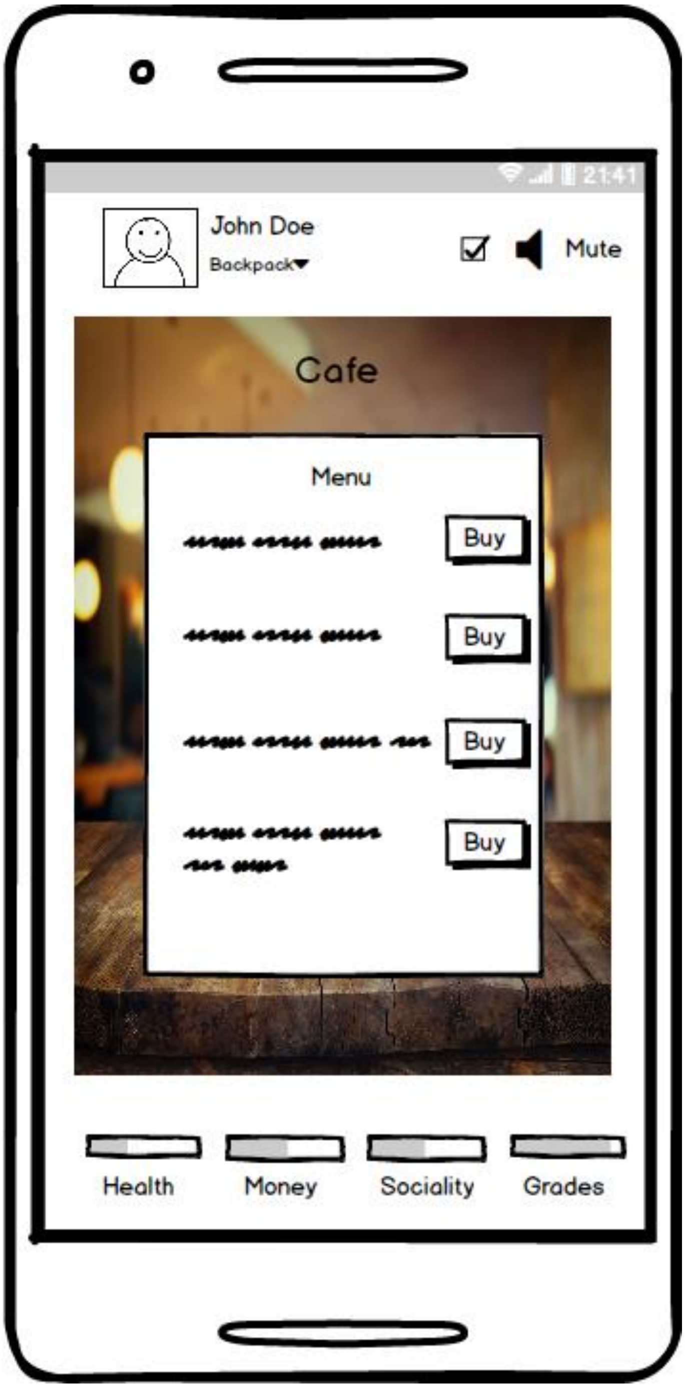


Figure: A Mock-up screening the "Cafe" scene of the Roomie

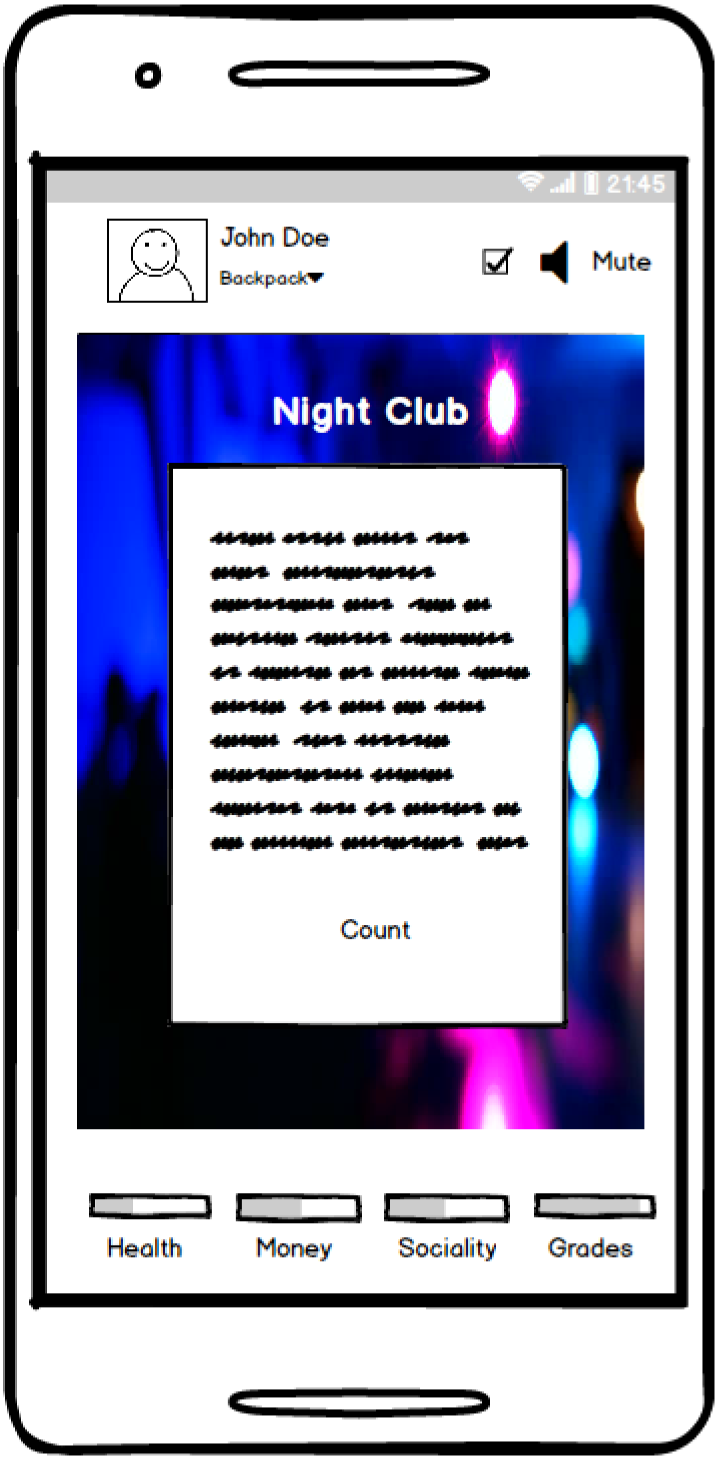


Figure: A Mock-up screening the "Night Club" scene of the Roomie

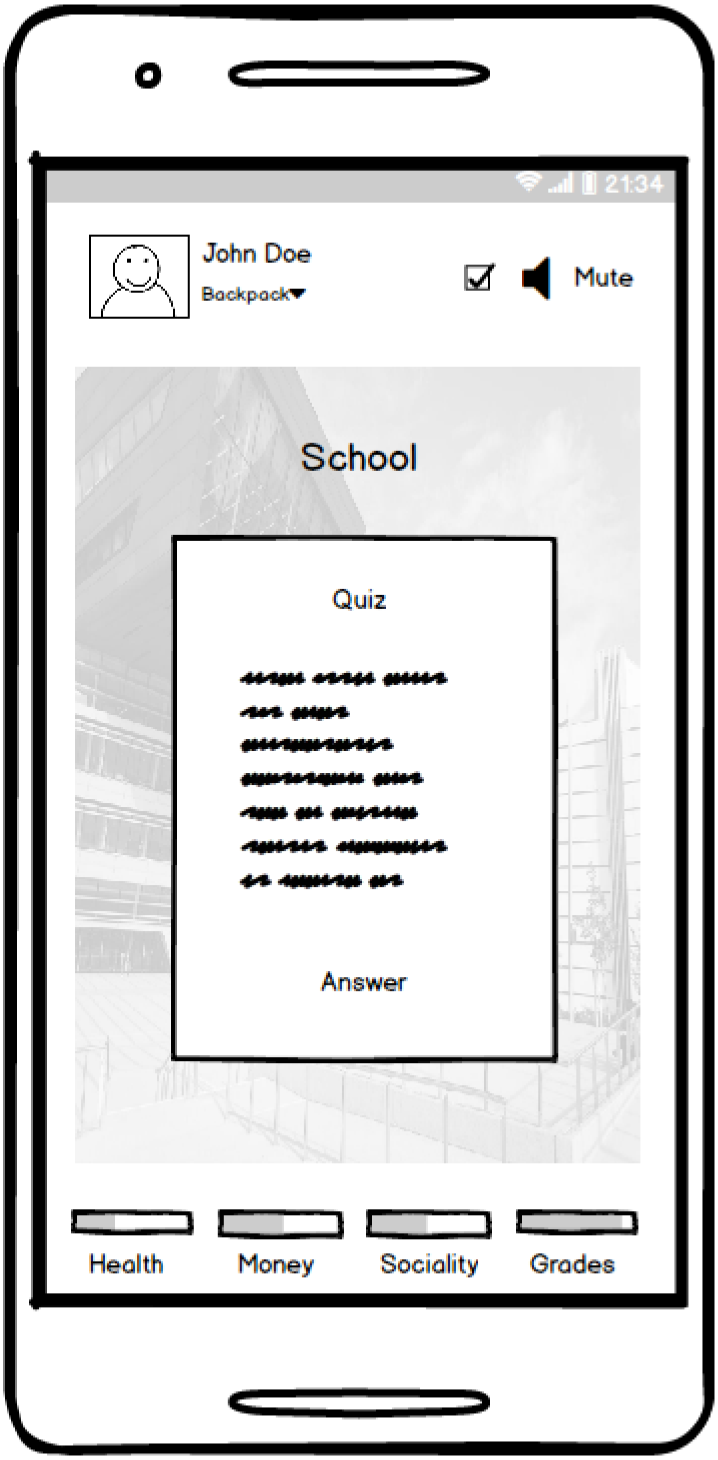


Figure: A Mock-up screening the "School" scene of the Roomie

2.1.3 Hardware Interfaces

All of the user inputs will be taken by the touchscreen. In addition to this, text area inputs will be taken by the keyboard that Android system automatically utilizes. The sound level of Roomie could be arranged by using volume buttons of the device. The other buttons on the device will only change the stage of the activity's lifecycle.[4]

2.1.4 Software Interfaces

- The game will be written in Java since Android applications developed using Java. Android API 23 is the developement level to support much more devices.
- IntelliJ Idea and Android Studio will be the development environments.
- For version control, the GitHub will be the main service. in addition, some pluggings and GitKraken are going to be used in order to handle the git operations more efficiently.

2.1.5 Memory Constraints

The program will use the default settings of the Android and JVM.

2.1.6 Operations

There are two main modes in Roomie, the first one is the "Game" mode, and the other one is the "Loading" mode. When the user opens the app, the game is in the "Loading" mode and the user is not expected to interact with the game. After loading the game, the "Game" mode starts. The touch-screen of the users' device is activated for taking user input. By making the appropriate touches to the screen, the user can proceed in the game.

The user can use the built-in buttons of his device to close the application or change the volume of the sound.

2.1.7 Site Adaptation Requirements

In the Roomie, initial data setups will be handled in the usual way. Application will use the android actions and device's Internal Storage. All initialized data which created by developers will be stored in the phone. When the application started, the Roomie will use the device storage firstly, to retrieve APK datas, secondly, retrieve the last saved game and lastly load the current saved game back in to the storage as the new save file. During the game play, the backgrounds, items' images, sounds and application theme is going to retrieved from the APK resources of the Roomie. All usable objects in the map will placed in fixed orientation. Therefore, the transaction of the objects will be same as the room change. Furthermore, all environments will be loaded just-in-time. During the item glow, the items will placed as buttons to the activity of the Roomie therefore the glow and the included animations will handled using the android's pre-existed animation and front-end libraries. All visual activities will be rendered to the game activity screen in real time.

In different occasions, Roomie will require additional functionality for mission related issues but these functions will not affect the front-end of the game. These functions are activating and collecting data from the Gyroscope and microphone sepera

2.2 Product Functions

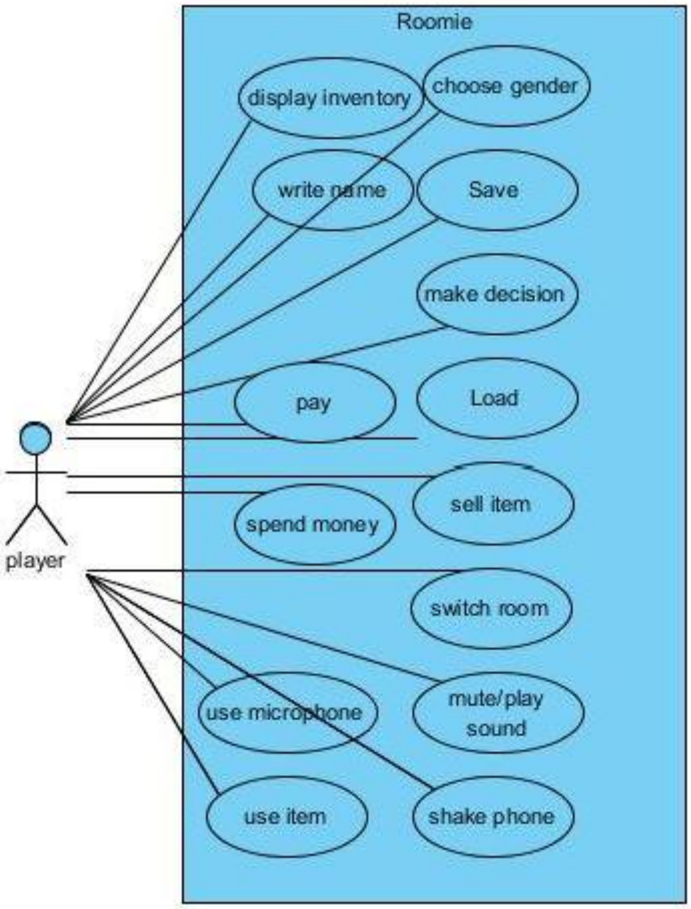


Figure 7 Use Case Diagram

At the very beginning of Roomie, the player needs to indicate his/her name and choose a gender. While playing Roomie, the player could switch between rooms, make decisions while answering the prompted questions, use items to improve his/her stats. For completing particular missions, the player might be required to shake their phone and use his/her microphone. Roomie is based on a series of events which consist of missions and questions. When the events are completed, the player may gain random items. These items are going to be displayed in their ‘backpack’. Using an item hub, the player will have the ability to sell or use the items in their ‘backpack’. During the game, Roomie will offer the players a set of different environments, diversifying the missions and questions. In the student flat, the player is able to navigate between the rooms. When Player closes or minimizes the Roomie Android Task, the game data will be automatically saved. Subsequently when the player reopens the game, Roomie will continue from its previously saved data without any progress loss.

- Write Name: This case is allowed when the player chooses a name , the chosen name will be the same throughout the entire game and it will be used to address the player.
- Choose Gender: This case is allowed when the player chooses a gender, the chose gender will be the throughout the entire game and it will be used to offer a more appropriate story-line.
- Switch Room: This case is allowed when the player is in his/her flat to navigate between the rooms.
- Make Decisions: There will be questions and missions in the game and the players are expected to make choices while answering those questions and completing the missions.
- Use Item: There will be randomly given items at the end of some events that the users are able to collect, the players can use these items to boost their stats bar levels.

- Sell Item: The players can sell items from their backpackir phone and trigger the motion sensors in order to complete some missions that are assigned during the game.
- Use Microphone: The players might need to use the microphone of their phone and trigger the motion sensors in order to com if they wish using Item Hub.
- Shake Phone: The players might need to shake the complete some missions that are assigned during the game.
- Pay: The players can pay rent with money.
- Spend Money: The player can spend money for different situations.
- Display Inventory: The players are able to to access their inventory (the items that they have collected throughout the game) using Item Hub.
- Mute/Play Sound: There will be some unique musics and sound effects for Roomie that is played in the background. The player is able to mute/play the music.
- Save: When the player closes Roomie Application, the game will be automatically saved.
- Load: When the player reopens Roomie, the game will automatically load from the previously saved game data.

2.3 Use Case Scenarios

When the user finishes to choose a gender and a name of the Roomie character, the game is going to start with the game state. In this state, the screen will display the main game page and the initial game environment which is home.

To start with the Roomie's story line, the user has to initiate the first event. With the current event, Player have to make the first choice.

2.3.1 Story Scenarios

In order to complete the various scenarios, Player have to fulfill the events and depend on the Player's decision, the game will move to another story part. To complete the events, Player have to evaluate the given situation carefully, and choose the decision that is provided by the Roomie. In order to choose, the user simply clicks to the option which is popped up to the screen. Then, consequently, Player may face another story piece, die, finish the game or extreme case which is more complex and full of dilemmas.

2.3.1.1 House Scenarios

In the Roomie House, Player has the ability to walk between the rooms and do various events.

Each room is going to be equipped with unique items. These items will be the one of the main foundation stones of our story line. Furthermore, these unique items have the ability to initiate an event. If the item is linked to an event, it will glow in order to appeal more easy to eye. These glowing items are going to be clickable, if user wanted to do the item's event, Player is required to tap to the item in order to start it.

In the Roomie House environment, the following rooms will provided the sample events:

- Living Room: This room is the main playground. A table and a game console is available to provide social, academic and health related circumstances to the player which may result changes in the player's statuses.
- Kitchen: In the kitchen, basically the Player have to clean the leftovers and the dishes in order to maintain the health status well.
- Bathroom: In the Bathroom, Player have to take care of needs. These events will also be related to the health status.
- Bedroom: In the Bedroom, the user is able to sleep which results in health improvement.

2.3.1.2 Outdoor Scenarios

In addition to the house environment, Roomie is also provide 4 different outdoor environment.

These are:

- University Classroom: Player has to solve the quizzes which will generate academic points.
- Library: In the library, as general assumption, Player has to be quiet during the study time. To do this, Roomie is going to access the user's telephone's microphone and decides whether Player is good student or not.
- Cafe : In the Roomie Cafe, the user is expected to socialize. To do this, he must choose meals which listed as different in prices. The more the Player spends, the more social he gets.
- Night Club: In the night club, the Player has to dance in Roomie in order to gain social points. To do this, player has to initiate the motion sensors, for instance, activations like shaking the phone is a trigger for the motion, therefore the user is assumed to complete the dance mission.

2.3.2 Active Event Scenarios

Roomie will have different kind of events in order to prevent the monotonic game play. These cases are dance, quizes and library missions. Some of these missions will use Android properties such as motion sensors and microphones, others will be depend on the Player's intelligence such as missions like quizes.

2.3.3 Die Scenarios

Without completing the whole flow chart of Roomie, the player might die after making some careless decisions. Since the flow of events are designed to create a unique life simulation, player may lose all of his health points with one of those unique paths.

In addition to this, throughout the game, user is always required to take care of his health points therefore he must eat/drink and take care of his needs.

2.3.4 Win Scenarios

The win case in Roomie is to complete a whole path until the end. For example, if a player manages to go all way through the game while making his decisions and avoiding accidentally dying, at the end he could graduate from the university and will be presented with a future-telling about his life.

2.3.5 Extreme Scenarios

In the game, depending on the events, there might be generated extreme events to surprise the Player. In the extreme events, the prize will be highly rewarding in terms of statuses' points whereas the deadly cases are also much more probable.

2.3.5 Expense Scenarios

In Roomie, the player will face with many situations that has to pay some expenses. For example, the game itself will generate household expenses such as rent and bills. These qualities will help Roomie to resemble a real-life experience more.

2.3.6 Randomized Item Drop Scenarios

After completing an event successfully, the Roomie is going to distribute an item to the player as a token of success. These items are predefined by the system but the possibility of dropping them differs based on the player's choices. Each item is going to developed uniquely such that each item will have different contributions to the player.

2.3.7 Status Boost Scenarios

The items that the player collects at his backpack can be used to boost status levels. Each item has a unique status points. For example, wearing "sunglasses" item while going to school will gain the user some amount of sociality points. Another example would be eating the "fresh sandwich" item which will result in some health points.

2.3.8 Status Bar Changing "Prize" Scenarios

While making choices, the player may increase/decrease his status points. For example, if the player manages to solve a quiz question in the class therefore he gains academic points. If the player chooses to eat some healthy food in the Cafe, it will make him gain health points. However, if he misses a class because he chooses to sleep instead, he will lose academic points but might gain health points. The player might also earn money at the end of some events.

2.4 User Characteristics

Roomie is designed for the Android platform therefore the users are not expected to have a background knowledge about technical issues. Only basic knowledge about how to use an Android device should be sufficient. The game is especially designed to have an uncomplicated graphical user-interface to provide users a fun and simple game experience.

2.5 Constraints

Developers must upload Roomie with up-to-date Java and minimum of Android API Level 23. The lower versions of Android s not supported by Roomie.

2.6 Assumptions and Dependencies

As developers, we are designing Roomie for the Android smart-phones which are expected to include a multi-colour touch screen, a proper set of motion sensors, and a decent microphone. The screen of these devices are expected to be user-friendly and large enough to provide a nice experience of Roomie.

To avoid and resolve conflict issues, Roomie will be developed with API Level 23 and the up-to-date Java SDK.

2.7 Apportioning the Requirements

Since motion sensors and microphone provide information from the current Android OS Level of the phone, if Google changes the API libraries for Sensors, requirements and development stage may differ.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

Roomie is going to start on an Android screen with its initial boot page . The prior input to Roomie will be the touch motion of the user which is listened by the screen to initiate actions. Additional inputs will be used for required cases, for instance the gravitational motion which readed by motion sensors of the phone will be used in cases like shaking the phone to dance or collection sound data to complete missions that contains sound inputs. The sensors will be activated only when its required. Therefore, all sensors will not be active fully during the play time.

3.1.2 Hardware Interfaces

Roomie is going to use the Android phone and sensors provided within. For primary input, the touches to the screen will be used, furthermore, sensor data will be handled and collected by built-in sensors of the phone. Roomie doesn’t require any additional sensors, also called 3rd party sensors which connected externally to the phone. Sensors will be activated during the special events. All the inputs are processed and collected by Roomie. In order to handle the sound, the built in speaker will be used for it. All transactions will be handled real time.

3.2 Functional Requirements

The roomie is only contain three functional action. Following list contains the actions and their functions.

- Touch
  - Touch motion will be the main input. With touching the gloving objects, the game will create events, the choices will be determined by touching the options, etc. Every interaction will require a touch input.
- Sound
  - In some cases, user have to use the microphone in order to complete relevant tasks which uses sound collection. For instance, in library, user have to be silent for 15 seconds and in order to pass the mission, Player must be silent and this transaction will be understanded by microphone of the smart phone
- Motion
  - For particular missions, user have to use motion sensors in order to complete gyro-missions. For example, in the dance club, Roomie dictates that user have to dance for a particular amount of time, therefore, to simulate this action, triggering the motion sensors by shaking the phone will be required to complete the mission successfully.

3.3 Performance Requirements

- The game consists of an Android application. Roomie can be played on only Android supporting phones with a minimum of API 23 level. There is no outside communication built-in to the application.
- The game supports only one player actively since it's designed as a single player game.
- Touch Screen and its inputs will used in order to change the state of the game. For instance the actions, events, items etc.
- All changes will be displayed on the main game screen.
- Microphone input will be used to determine the success of given tasks, and also motion sensors will be used to detect motions.

- Taptic sensors will be used in some extreme cases.
- Internal Storage Device (Internal SD) will be used in order to save the data of the Roomie.
- All Data transaction of the game will be done by Roomie in order to reduce dependency and operation time.

3.4 Design Constraints

Our design will be limited by Java and Android for GUI, and other limitations will be imposed by capabilities of Android’s SensorListener and AudioRecord.

3.5 Software Design Attributes

3.5.1 Reliability

The user should have an Android device since Roomie will run on that and the reliability of the system is dependent to the reliability of Android.

3.5.2 Availability

Since Android API 23 (Marshmallow) is going to be used in the project developement, Roomie will be available to those with Android version 6.0 and above.

3.5.3 Maintainability

To keep our software’s maintainability at a high level, we plan to keep our code clean and documented. We are planning to limit the number of external libraries used in the project. In addition to these, we are planning to avoid writing complicated code, and keep the code as simple as possible while being organized. Another approach we are going to take is testing the code in various devices to assure we do not encounter any problems.

3.5.4 Portability

The system is portable since it will run on any Android device with Android version 6.0 and above.

3.5.6 Extendibility

We plan to make Roomie suitable to be reused and extended for further versions and other works.

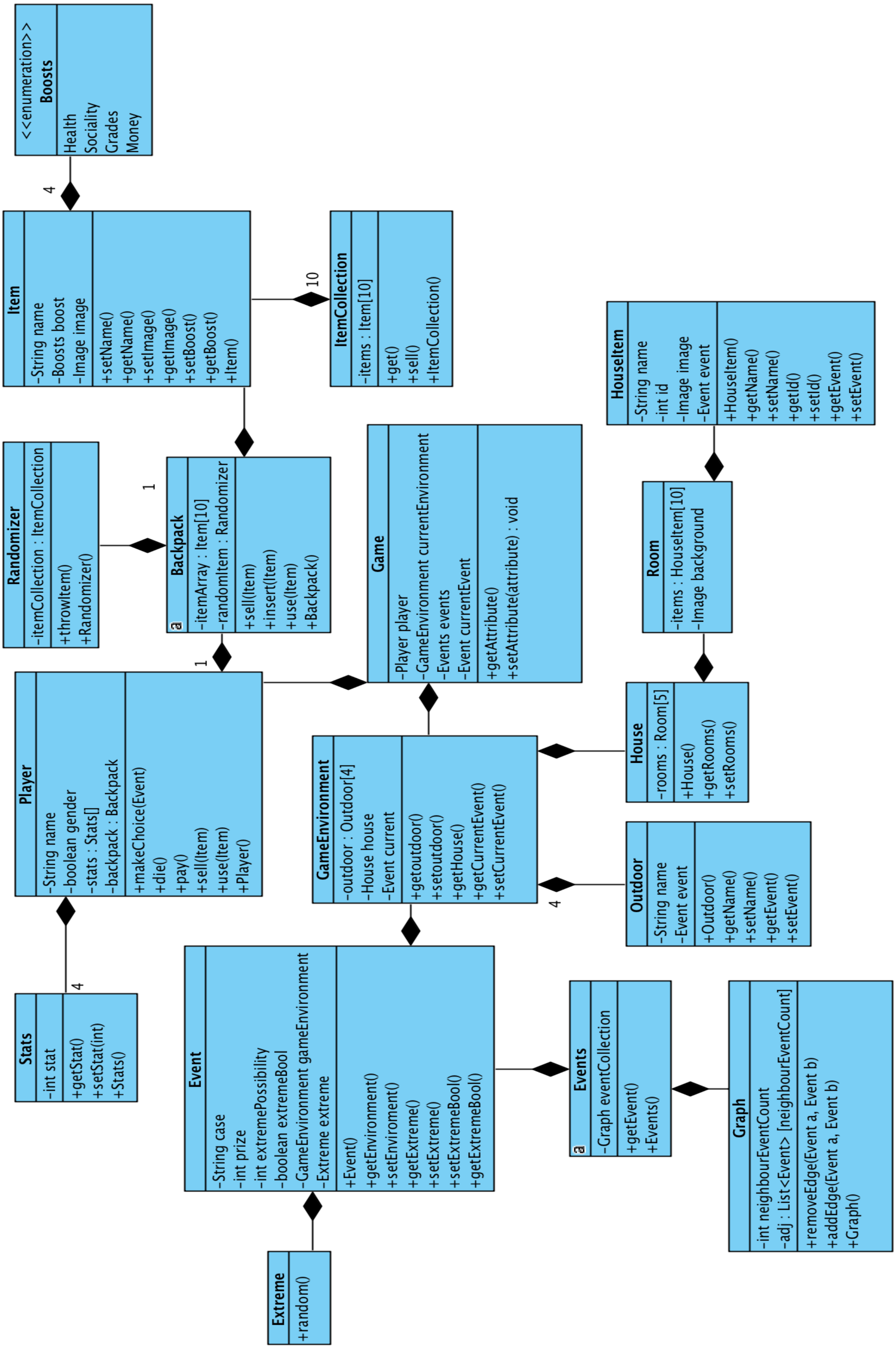
3.5.7 User-friendliness

We plan to build our application as user friendly as possible. Our first aim is to make the user interface easy to use. We plan to apply a minimal color range to our design. We will use icons to guide the users and use iconography to some actions.

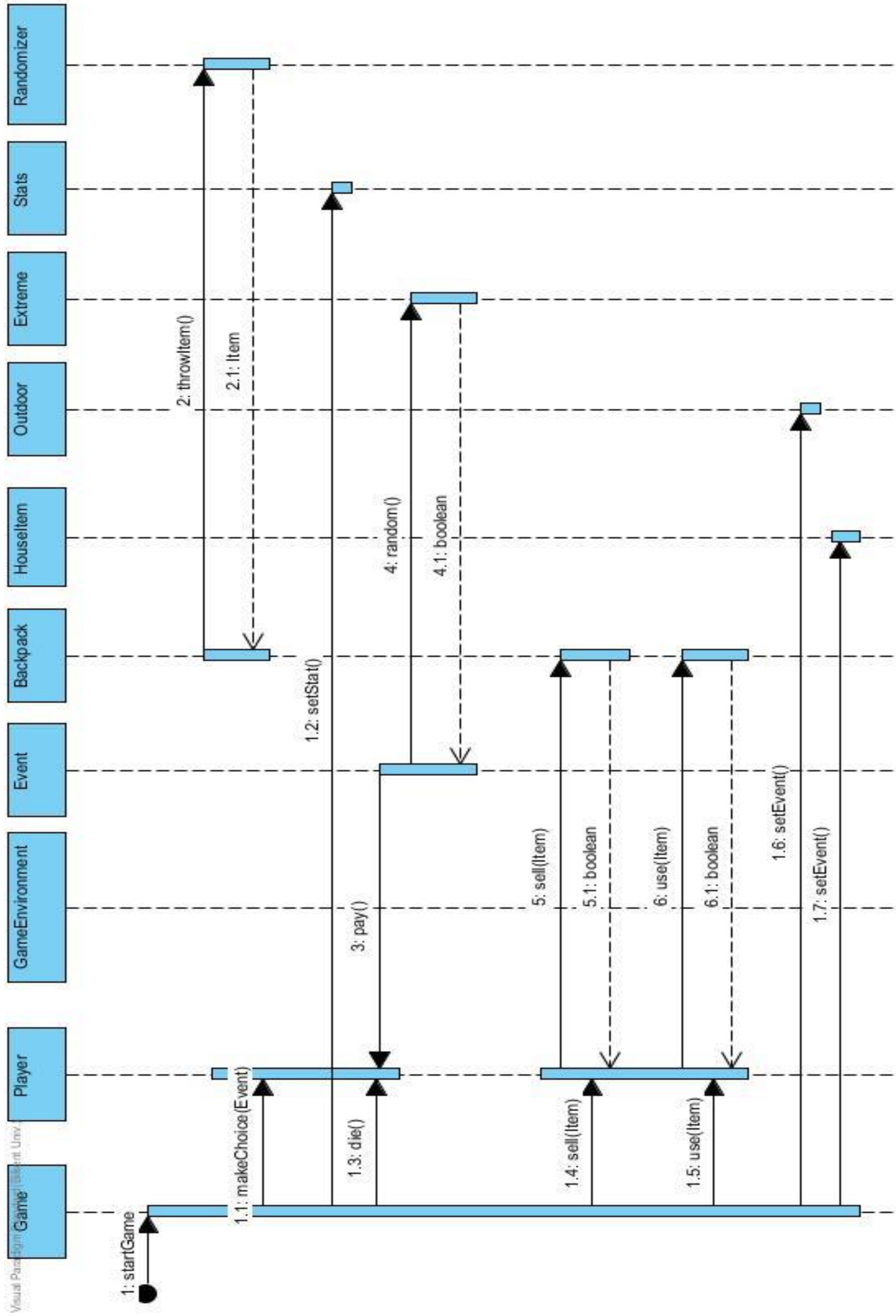
4. Appendixes

Here is the class and sequence diagram of Roomie that shows the main objects and functionality of those objects:

- Figure A: Class Diagram of the Roomie
- Figure B: Sequence Diagram of the Roomie



Appendix Figure A



Appendix Figure B

5. References

[1]-Motion Sensors. (2017, August 08). Retrieved October 07, 2017, from https://developer.android.com/guide/topics/sensors/sensors_motion.html

[2]-AudioRecord. (2017, July 24). Retrieved October 07, 2017, from https://developer.android.com/reference/android/media/AudioRecord.html

[3]-Android 6.0 APIs. (2016, October 07). Retrieved October 07, 2017, from https://developer.android.com/about/versions/marshmallow/android-6.0.html

[4]-The Activity Lifecycle (2017, September 26). Retrieved October 07, 2017, from https://developer.android.com/guide/components/activities/activity-lifecycle.html

[5]- Activity. (2017, July 24). Retrieved October 07, 2017, from https://developer.android.com/reference/android/app/Activity.html