Software Engineering

Report

Love Letter

Ibrahimkhanli Murad

Imanzade Farid

Rahmanova Ofelya

Sideif-zada Saida

May 22, 2020

Abstract

This report is a summary of the "Love Letter" game for Software Engineering classes at French-Azerbaijani University (UFAZ). This project is intended to put into practice the knowledge and techniques acquired in the course of Software Engineering. The task is to propose an implementation of "Love Letter" - card game which is played by 2 to 4 players. In this report, a brief information about software engineering topics, game manual and our progress is given, and UML diagrams along with Agile reports are demonstrated.

Contents

List of Figures

Introduction

Purpose

This project is intended to put into practice the knowledge and techniques acquired in the course of Software Engineering. Project allow us to experiment with new topics, techniques and tools. The purpose of this project is to propose an implementation of "Love Letter" - card game which is played by 2 to 4 players. The rules that we respected during implementation phase are described in "Design & Specification" stage(Look at 2).

Objectives

- Practice and improve ourselves on various areas of software development
- Gain skills in developing effective software
- Practice with Agile methods
- Develop effective teamwork skills
- Propose an implementation of Love Letter - card game
- Demonstrate as much as possible the qualities of a good Man-Machine Interface
- Localize interface for at least two countries speaking different languages
- Form a coherent "whole" User Interface
- Implement Artificial Intelligence module for game
- Make application responsive for mobile
- Implement continues integration and deployment
- Implement unit tests
- Write detailed and quality report
- Propose user manual and graphs

Descriptions

Software Project : Software engineering is an engineering branch associated with development of software product using well-defined scientific principles, methods and procedures. The outcome of software engineering is an efficient and reliable software product.
A software development project is a complex undertaking by several persons within the boundaries and staff resources that produces new or enhanced computer code.
Software project management is essential to incorporate user requirements along with budget and time constraints. The image on the right shows triple restrictions for software projects. It is an essential part of software organization to deliver **quality** product, keeping the **cost** within client's budget constrain and deliver the project **in time**.

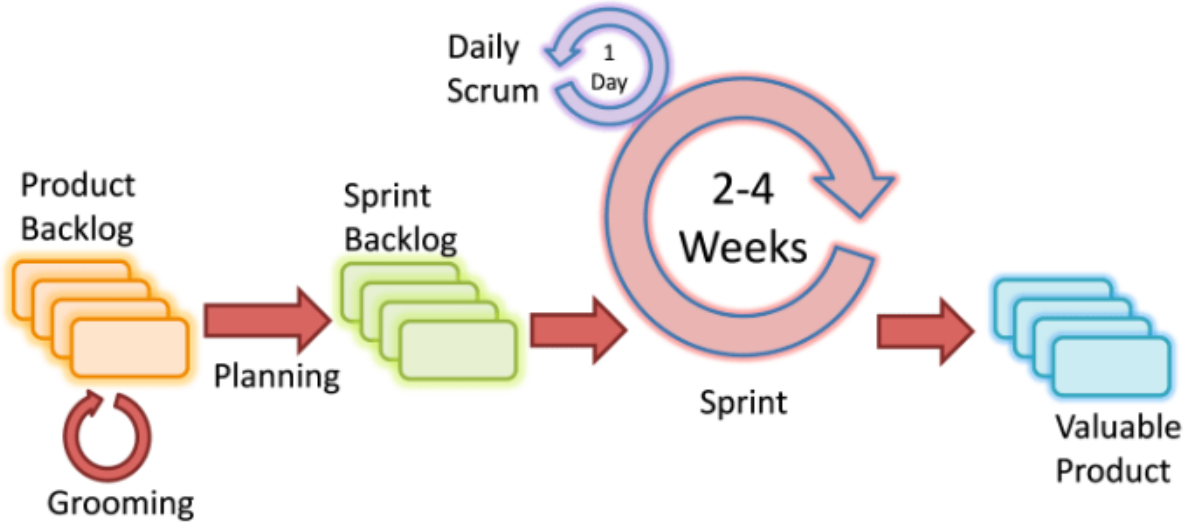
A Software Project is the complete procedure of software development from requirement gathering to testing and maintenance, carried out according to the execution

methodologies, in a specified period of time to achieve intended software product.
During the lifecycle of a project, some particular entities can be distinguished:

- The **customer** is the one who **expresses the need**.
- The **supplier** is the one who **meets the need**.
- The **stakeholder** is the one who has any interest in project's outcome. Other than customer and supplier, it is typically members of a project team, project managers, executives, project sponsors, and users.

Software Development Life Cycle Model : SDLC model - is a series of phases that provide a common understanding of the software building process. How the software will be realized and developed, and requirements elicitation phase to convert these ideas and requirements into functions and features until its usage and operation to achieve the needs.
It is important to choose the right SDLC model according to the specific concerns and requirements of the project to ensure its success.

Agile Model : The agile method involves breaking down project into prioritized requirements, and delivering each individually within an iterative cycle. An iteration is the routine of developing small sections of a project at a time. Each iteration is reviewed and assessed by the development team. The insights gained from the assessment are used to determine the next step in development. Detailed goals are set in each iteration meeting such as: expected changes, time estimates, priorities. Agile project methodology breaks down projects into small pieces that are completed in work sessions or sprints that run from the design phase to testing. Sprints are generally short, running over days or weeks.



Unit Testing : UT - is the first level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. Unit Testing is performed prior to Integration Testing.

Continuous Integration : CI - s the practice of automating the integration of code changes from multiple contributors into a single software project. By integrating regularly, it is possible to detect errors quickly, and locate them more easily. Additionally, Continuous Deployment and Continuous Delivery have developed as best-practices for keeping your application deployable at any point or even pushing your main codebase automatically into production whenever new changes are brought into it. This allows your team to move fast while keeping high quality standards that can be checked automatically.

Project Life Cycle : A project life cycle is the sequence of phases that a project goes through from its initiation to its closure.

Needs Analysis:

Data collecting. Redefining real needs by understanding the context and analyzing needs expressed by the client.

Specification:

Describing previously analyzed needs with more details in the form of requirements that the solution must imperatively satisfy.

Design:

Providing the solution that meet all the requirements identified during specification phase.

Implementation:

Implementing the solution conceived during design phase.

Testing:

Testing the software, to improve its quality and to know the degree of presence of a particular quality.

Deployment:

Delivering the software into production, making it available to the end user.

Maintenance::

Modifying the current software, adding new functionalities, preventing anomalies.

Design & Specification

User Manual

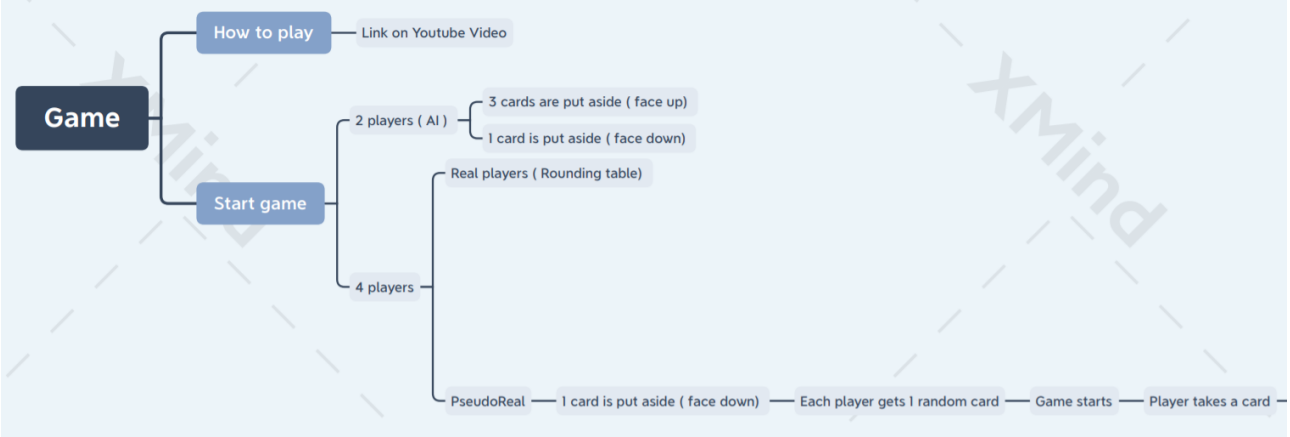


Figure 2.1: Game Model

How it works?

As the game app launches player can choose the preferred interface language **(English, French, Azerbaijani or Russian)**. If player does not know anything about the rules, he can click "How to play?" button and watch the video to learn and then click "Start Game". Otherwise, he can start the game directly.
After clicking "Start Game", player can choose the number of players (from 2 to 4) and their levels (easy or medium). For now, only easy level is available. Medium will be ready soon. When player clicks "Launch the game" button, cards are distributed to the number of players selected. If game starts with 2 players, then **1 unknown and 3 known cards** are reserved. In other cases, **1 unknown card is reserved**.
Player can click on "Show all cards" button to become more familiar with each card's abilities. As player clicks on "Start your turn" button, he gets randomly distributed card from card deck. Each player has **"brain"** property and during each round players' actions are being written to the "brain" and these actions are considered in the players' next turns. With respect to the game rules, as the "brain" compares the previously distributed players' cards, it decides which cards will be distributed next for the game to be logically correct and interesting.
The game proposes the following scenario and rules. For example:

- If player chose to play with "Guard" card, he must choose the player and try to predict his card. If the prediction is correct, chosen player loses the round.
- Choosing to play the "Priest" card gives the player the ability to look at other player's card for 3 seconds.
- As player chooses the "Baron" card, he can compare his cards with the other player. The player, with lower card value loses the round. If the card values are the same, player with lower sum of values of discarded cards loses the round.
- Playing the "Handmaid" card gives player an advantage of ignoring other players actions until the next turn.
- Playing the "Prince" card, you can choose yourself or the other player to discard his card and draw another one.
- If player chooses the "King" card, he can choose the player to exchange his cards with him.
- If player has the "Countess" card along with "King" or "Prince" card, he has to discard the "Countess" card. If player will not do so, he loses the round.
- Getting the "Princess" card, player should not discard it. If player did so, he loses the round.

At the end of each turn, player can see the actions played by AI in a dashboard to be able to think about the further strategy of the game.
During the game, player can see the number of won rounds of each player.

UML Diagrams

Class Diagram

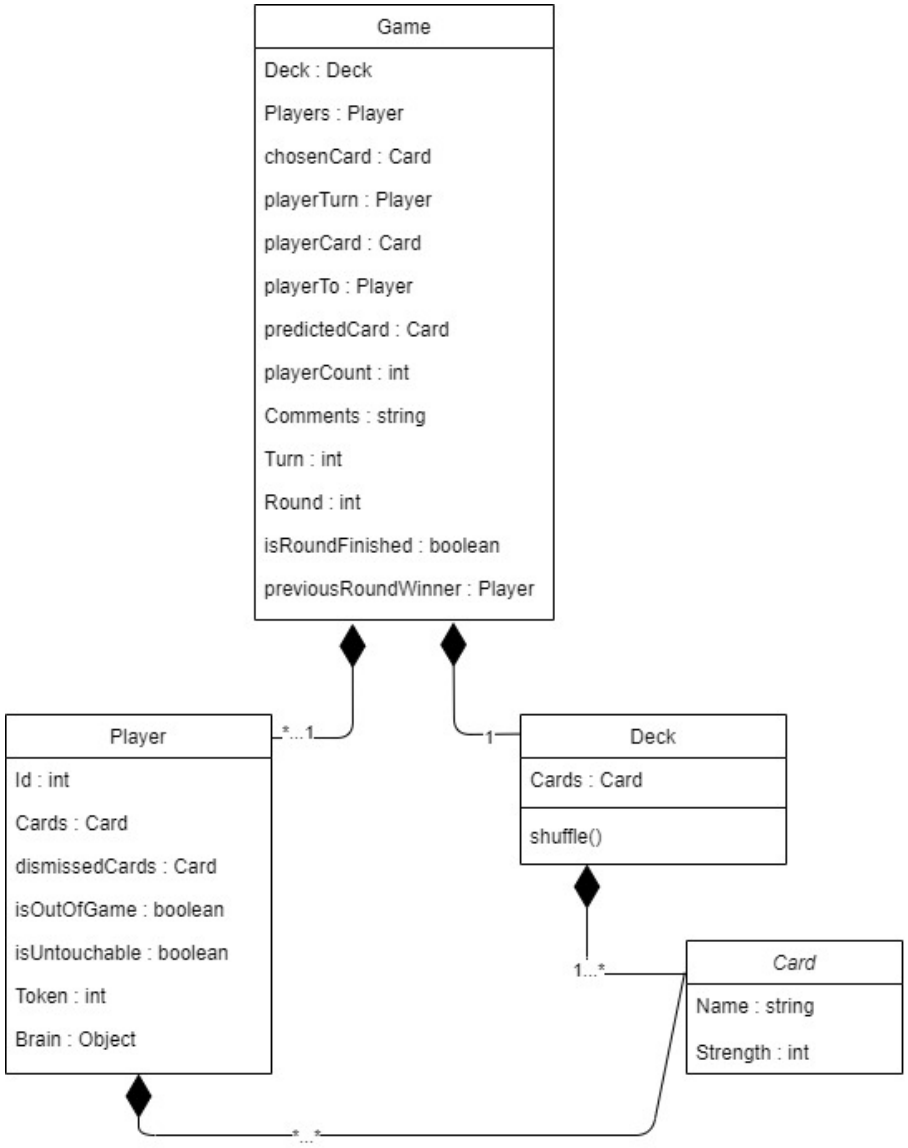


Figure 3.1: Class Diagram

Use-case Diagram

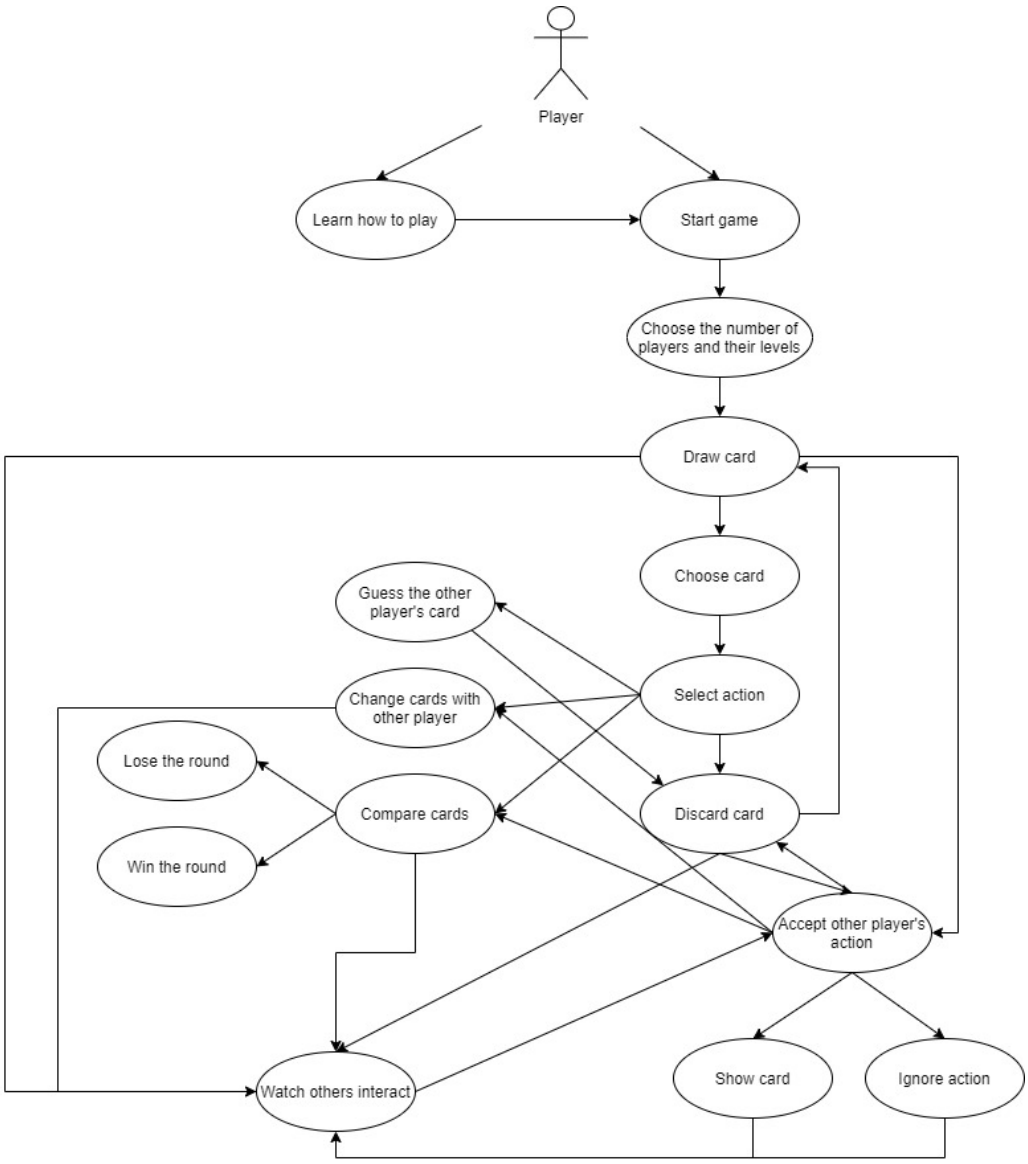
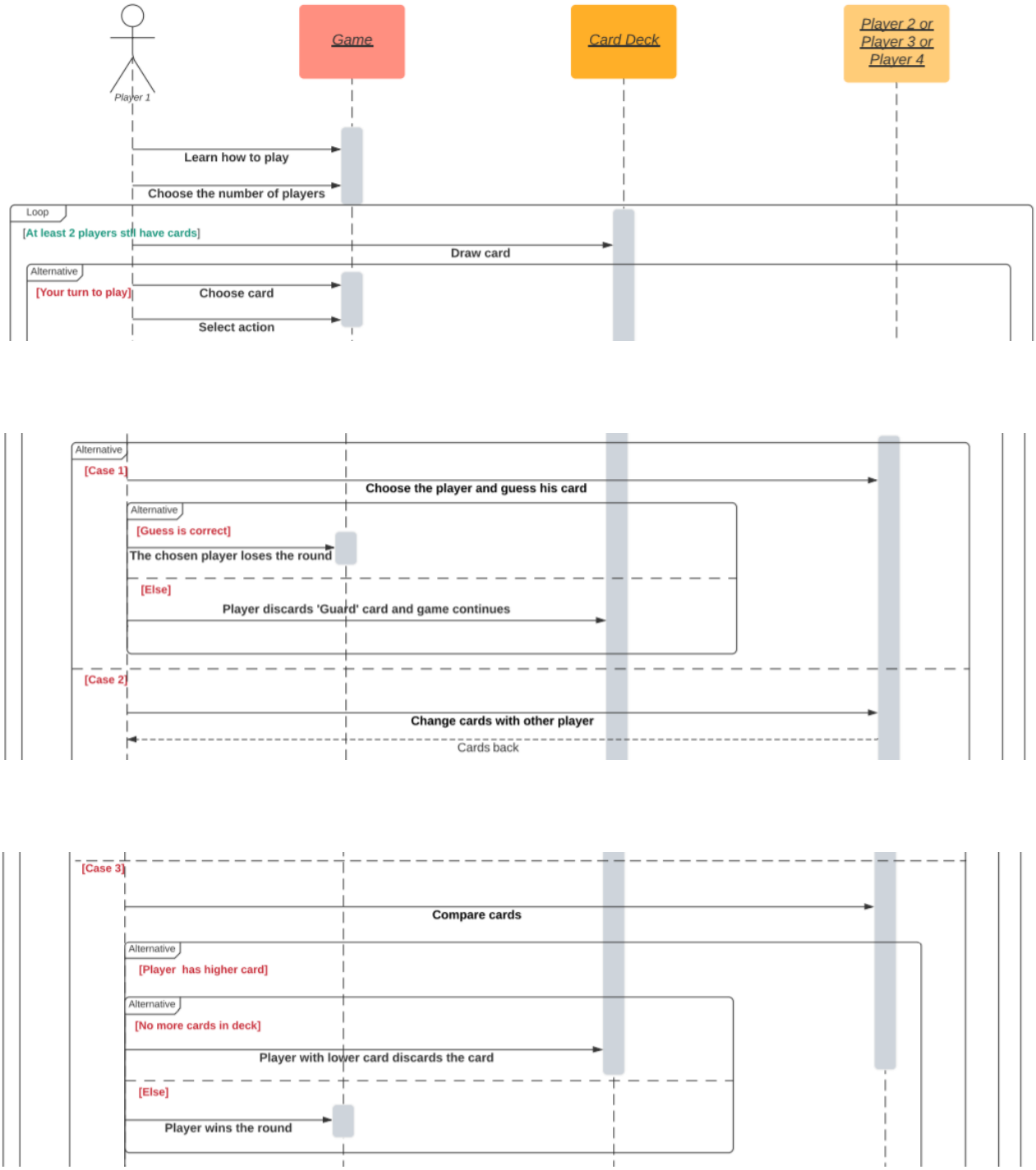


Figure 3.2: Use-case Diagram

Sequence Diagram



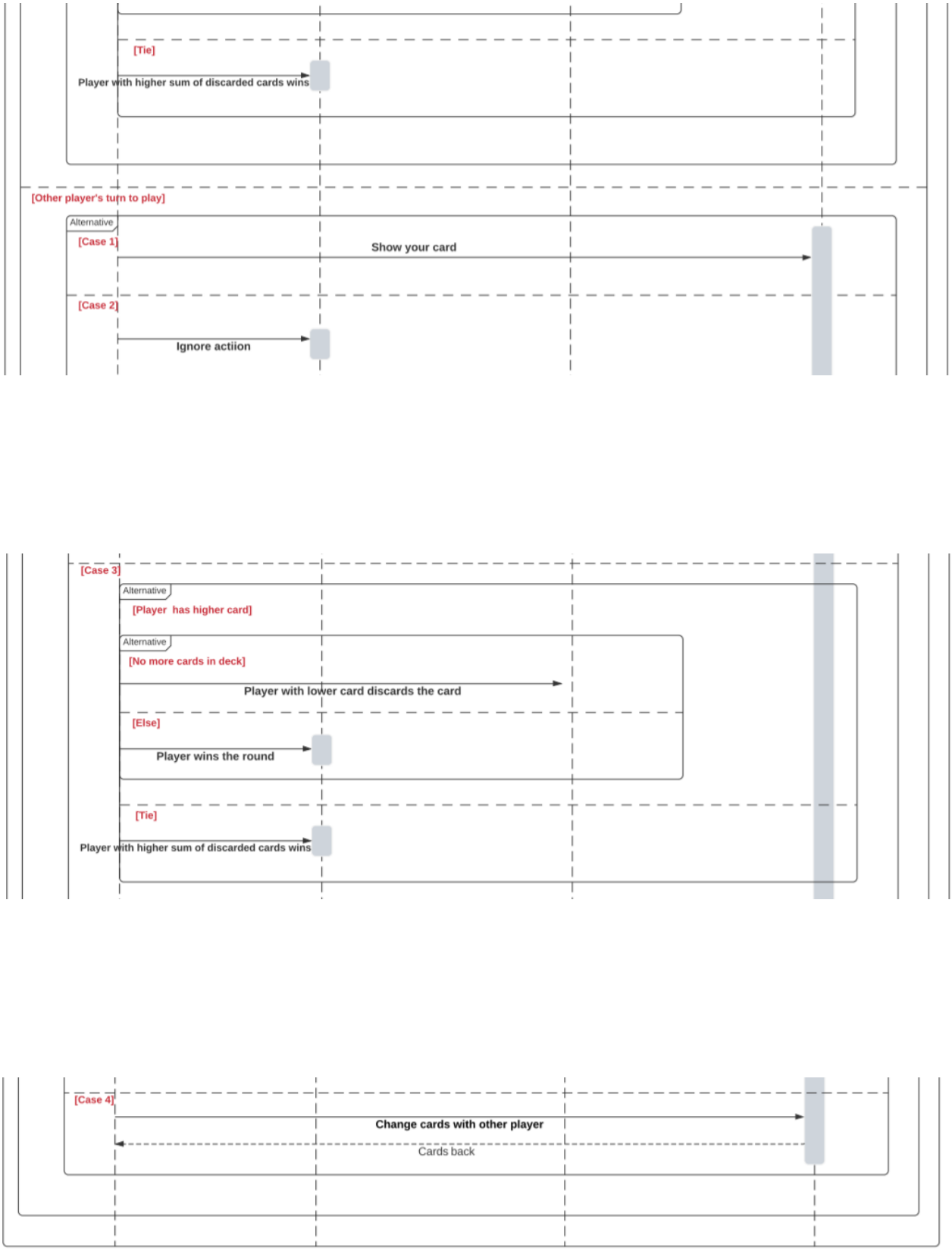


Figure 3.3: Sequence diagram

Team and Project Organization

Sprints

First, we divided the project into 5 sprints and were planning to start working on the project on the April 1. We were aiming to spend two weeks for each sprint. Unfortunately, due to some unpredicted time issues, the start date was delayed to April 20. Therefore, each sprint was reduced to one week time frame.

The project was divided into 5 following sprints: Project Initalization (April 23- May 30), Development Phase 1 (May 1-May 5), Development Phase 2 (May 6-May 11), Testing (May 12-May 17), Project closure (May 18-22).

1. Sprint 1. Project Initalization. Initial meetings, dicussioins, planning and task assignments.(Figure 4.1)

1–6 of **6** as at: **22/May/20 11:12 AM**

T	Key	Summary	Assignee	Reporter	P	Status	Resolution
☑	SP-7	Design general view of project	Ofelya Rahmanova	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-5	Divide roles between team members	Ibrahimkhanli Murad	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-4	Decide programming tools for project	Ibrahimkhanli Murad	Ibrahimkhanli Murad	↓	DONE	Done
☑	SP-3	Create roadmap of our process	saidanur.sideif-zada	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-2	Create bitbucket repository for project	Farid Imanzade	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-1	Create general map of game rules	saidanur.sideif-zada	Ibrahimkhanli Murad	↑	DONE	Done

Figure 4.1: Issues and tasks of Sprint 1 - Project Initialization

2. Sprint 2. Development Phase 1. Preparation of primary HTML skelethon and front, start of the coding process.(Figure 4.2)

1–15 of **15** as at: **22/May/20 8:34 PM**

T	Key	Summary	Assignee	Reporter	P	Status	Resolution
☑	SP-21	Write pseudo-code of 1 round of a game	saidanur.sideif-zada	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-20	Create design of game round view	Ofelya Rahmanova	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-19	Implement startGame and startTurn logic	Ibrahimkhanli Murad	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-18	Develop front-part of additional cards	Farid Imanzade	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-17	Design player selection modal	Ofelya Rahmanova	Ibrahimkhanli Murad	↓	DONE	Done
☑	SP-16	Break down website to small components	Ibrahimkhanli Murad	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-15	Add player selection modal	Farid Imanzade	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-14	Implement objects in js	Ibrahimkhanli Murad	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-13	Write down objects and parameters of game	Farid Imanzade	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-12	Develop html skeleton of website	Farid Imanzade	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-11	Implement pubsub logic in project	Ibrahimkhanli Murad	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-10	Implement state management logic in project	Ibrahimkhanli Murad	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-9	Configure initial project structure	Ibrahimkhanli Murad	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-8	Create design of a home view	Ofelya Rahmanova	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-6	Brain storm	*Unassigned*	Ibrahimkhanli Murad	↑	DONE	Done

Figure 4.2: Issues and tasks of Sprint 2 - Development Phase 1

3. Sprint 3. Development Phase 2. Implementation of back-end and game logic.(Figure 4.3)

1–18 of 18 as at: 22/May/20 8:32 PM

T	Key	Summary	Assignee	Reporter	P	Status	Resolution
☑	SP-48	Add rounds logic to game	Ibrahimkhanli Murad	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-45	Make responsive game screen	Ofelya Rahmanova	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-44	Make responsive screen for small devices	Ofelya Rahmanova	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-43	Add localization files to root folder	Farid Imanzade	Ibrahimkhanli Murad	↓	DONE	Done
☑	SP-42	Handle game till end if user is out of game, and finish round properly	Ibrahimkhanli Murad	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-39	Add comments to game logic	Ibrahimkhanli Murad	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-38	Add comments modal to page	Farid Imanzade	Ibrahimkhanli Murad	↓	DONE	Done
●	SP-32	Fix select option for prince choise, and add block(loading) for handymaid	Farid Imanzade	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-31	Add animations for actions	Farid Imanzade	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-30	Add actions for all cards	Ibrahimkhanli Murad	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-29	Handle user choise, make ready payloads and actions	Ibrahimkhanli Murad	Ibrahimkhanli Murad	↑	DONE	Done
●	SP-28	Fix python server mime configuration	Farid Imanzade	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-27	Develop small components	Farid Imanzade	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-26	Split up website to small changeable components	Farid Imanzade	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-25	Fill up in game objects	Ibrahimkhanli Murad	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-24	Create card payloads	Ibrahimkhanli Murad	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-23	Create card actions	Ibrahimkhanli Murad	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-22	Update object parameters	Farid Imanzade	Ibrahimkhanli Murad	↑	DONE	Done

Figure 4.3: Issues and tasks of Sprint 3 - Development Phase 2

4. Sprint 4. Testing. Testing the game, finalizing unfinished tasks from previous sprints, implementing unit tests.(Figure 4.4)

1–12 of 12 as at: 22/May/20 11:14 AM

T	Key	Summary	Assignee	Reporter	P	Status	Resolution
☑	SP-60	Implement unit tests for project using Jasmine	Ibrahimkhanli Murad	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-51	Add localization for entire project with stati18n(en, ru, az, fr)	Farid Imanzade	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-50	Implement round logic	Ibrahimkhanli Murad	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-49	Add round counter modal	Farid Imanzade	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-47	Separate reserved_cards components	Farid Imanzade	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-46	Fix game logic problems and bugs	Ibrahimkhanli Murad	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-41	Add support for 2 and 3 players	Farid Imanzade	Ibrahimkhanli Murad	↓	DONE	Done
●	SP-40	Fix small problems with game logic and design of front	Ibrahimkhanli Murad	Ibrahimkhanli Murad	↑	DONE	Done
●	SP-36	Fix conflict problems between branches	Ibrahimkhanli Murad	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-35	Create bitbucket pipelines for Continues Integration and Deployment	Ibrahimkhanli Murad	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-34	Setup heroku for deploying project	Ibrahimkhanli Murad	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-33	Choose cloud provider for deploying project	Ibrahimkhanli Murad	Ibrahimkhanli Murad	↑	DONE	Done

Figure 4.4: Issues and tasks of Sprint 4 - Testing

5. Sprint 5. Project closure. Finalizing the game and start of the project report.(Figure 4.5)

1–18 of 18 as at: 22/May/20 11:14 AM

T	Key	Summary	Assignee	Reporter	P	Status	Resolution
☑	SP-70	Final meeting	*Unassigned*	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-69	Write conclusion part of report	saidanur.sideif-zada	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-68	Write Agile part of report	Ibrahimkhanli Murad	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-67	Write UML diagrams part of report	Ofelya Rahmanova	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-66	Write technologies part of report	Ibrahimkhanli Murad	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-65	Write project guide part of report	Ofelya Rahmanova	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-64	Write sprint and roles part of report	Farid Imanzade	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-63	Write purpose part of report	saidanur.sideif-zada	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-62	Write description part of report	saidanur.sideif-zada	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-61	Create skeleton of report	saidanur.sideif-zada	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-59	Finish unit tests with Jasmine for entire project	Ibrahimkhanli Murad	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-58	Create Agile reports of project(burndown, velocity charts, sprint reports, pie charts)	Ibrahimkhanli Murad	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-57	Create UML diagrams for project(use-case, object, sequence diagrams)	Ofelya Rahmanova	Ibrahimkhanli Murad	↑	DONE	Done
■	SP-56	Fix all responsiveness problems	Ofelya Rahmanova	Ibrahimkhanli Murad	↑	DONE	Done
☑	SP-55	Update countess logic	Ibrahimkhanli Murad	Ibrahimkhanli Murad	↑	DONE	Done
■	SP-54	Fix priest show card problem	Farid Imanzade	Ibrahimkhanli Murad	↑	DONE	Done
■	SP-53	Fix preloader bug on Safari	Farid Imanzade	Ibrahimkhanli Murad	↑	DONE	Done
■	SP-52	Fix localization problems	Ibrahimkhanli Murad	Ibrahimkhanli Murad	↑	DONE	Done

Figure 4.5: Issues and tasks of Sprint 5 - Project Closure

Team member roles

Due to the short time limit, the tasks were assigned for each of us based on our skills in order to carry the workflow efficiently, smooth and speedily. We tried to ensure that our group is functional, that everyone can do their share of the work, in a way that suits their desires and abilities.

- Taking into account Murad's experience in back-end and devops, he was responsible with logic implementation and AI implementation, and setting up server for deployment. He was the one who distrubuted assignments and tasks among group members, carried out project report, set up and managed "Bitbucket" control repository hosting service. Murad cooperated in front-end and language localisation either.

- According to Farid's expertising field, he was assigned mainly front-end related tasks, such as, adding animations, visual effects, general appearance of the game, development of small components(dashboard, informative panels). He also contributed in setting up language localisation.

- Ofelia enhanced front-end by making the game responsive on all devices, such as laptops, tablets, mobile phones. She is the author of the full game design as well. UML diagrams and some tables were prepared by Ofelia that are in project report.

- Saida supported the project by contributing to game logic, providing necessary information for game and project development. Development of the project report was carried out mainly by Saida and Murad, partly by Farid and Ofelia.

Technologies used

There are optional and mandatory tasks for this project. We had to choose to implement either artificial intelligence or networking, or to propose a solution to allow several humans to play locally, on the same screen. All our team members has a lot of interest to Artificial Intelligence, but we could never try ourselves in this area. And also we had a lot of practice with networking on the past, so we decided to go with a new style. Because of time restriction, and a little experience on back-end between team members, we decided to avoid using back-end. So we came to decision to render all project on client-side, without using back-end and database. The only restriction is that, all data is cleaned up on page refresh.

Tools

- HTML5

  HTML is used for structuring and presenting content on browser. HTML5 was the fifth and last major version of HTML that is a World Wide Web Consortium (W3C) recommendation.

- CSS3

  CSS is used as a style sheet language used for describing the presentation of a document. CSS3 is the latest evolution of the Cascading Style Sheets language.

- jQuery3.5

  jQuery is a JavaScript library which we used to simplify HTML DOM tree traversal and manipulation, as well as event handling, CSS animation, and Ajax.

- JS

  JavaScript is the most popular implementation of the ECMAScript Standard. The core features of Javascript are based on the ECMAScript standard, but Javascript also has other additional features that are not in the ECMA specifications/standard. We used Vanilla JS along with jQuery in order to manipulate HTML DOM tree.

- EcmaScript

  ECMAScript(or ES) is a scripting-language specification standardized by Ecma International. It was created to standardize JavaScript to help foster multiple independent implementations. ES modules bring an official, standardized module system to JavaScript. ES module files are loaded using the standard Same-Origin policy restrictions that browsers enforce and have many other security restrictions in place, while JavaScript "script" files have much more lax security to avoid breaking existing websites as better security standards have been added to browsers over time.

  We built custom state management system and JS library in order to easily build our user interface.

- Python3

  Python is used to in order to set up HTTP server for our EcmaScript project,

- Git / BitBucket

Git is used as distributed version control system during software development lifecyle. Bitbucket is used as a repository hosting for our software. There are several good repository hosting services, but we decided to go with Atlassian products, so that we can easily connect our repository to our project management software. Besides that, Bitbucket much more simple setup, easily configured pipelines for continues integration and deployment and nice ui and different tools which makes version control much more easier.

- Jira

  There are a lot of product management tools on the market. But with current experience, we decided that Jira is the most suitable for Agile Project Management.

- Heroku

  Heroku is a platform as a service (PaaS) that enables developers to build, run, and operate applications entirely in the cloud. We decided to go with Heroku because it gives free plan and it is very easy to configure. It also provides you with built-in CLI and ssh access.

Issues

- Game is not supported on some versions of Safari, as old IOS versions doesn't have EcmaScript support.
- Because of browser rendering, there can be some problems if game is played quickly, please consider waiting for a second between actions,

Agile data

Velocity Chart

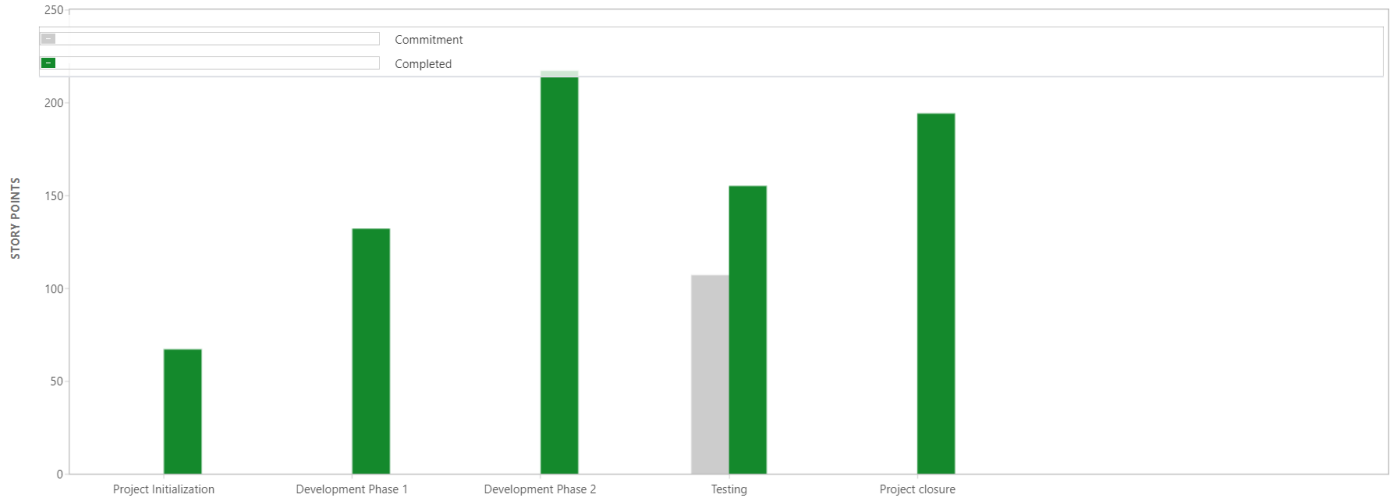


Figure 6.1: Velocity Chart - Part 1.

Sprint	Commitment	Completed
Project Initialization	0	67
Development Phase 1	0	132
Development Phase 2	0	217
Testing	107	155
Project closure	0	194

Figure 6.2: Velocity Chart - Part 2.

Person Data

Pie Chart Report

Project: Se Project (Assignee)

Chart

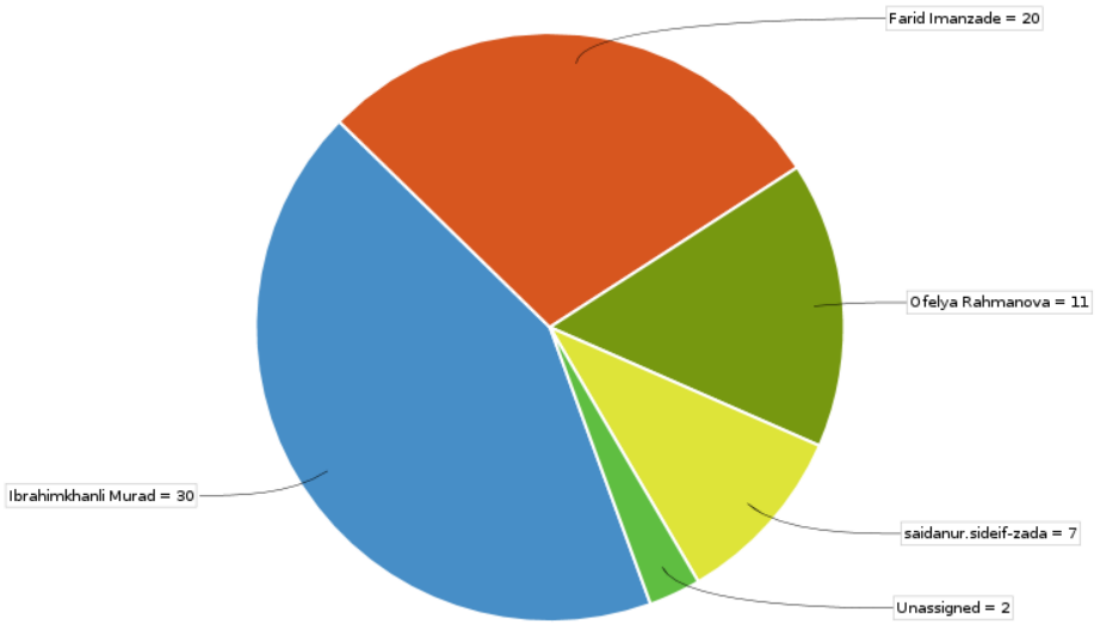


Figure 6.3: Assignee - Pie Chart.

Data Table

	Issues	%
Ibrahimkhanli Murad	30	42%
Farid Imanzade	20	28%
Ofelya Rahmanova	11	15%
saidanur.sideif-zada	7	10%
Unassigned	2	2%

Figure 6.4: Assignee - Table.

Components Data

Priority Data

Labels Data

Sprint Data

Pie Chart Report

Project: Se Project (Components)

Chart

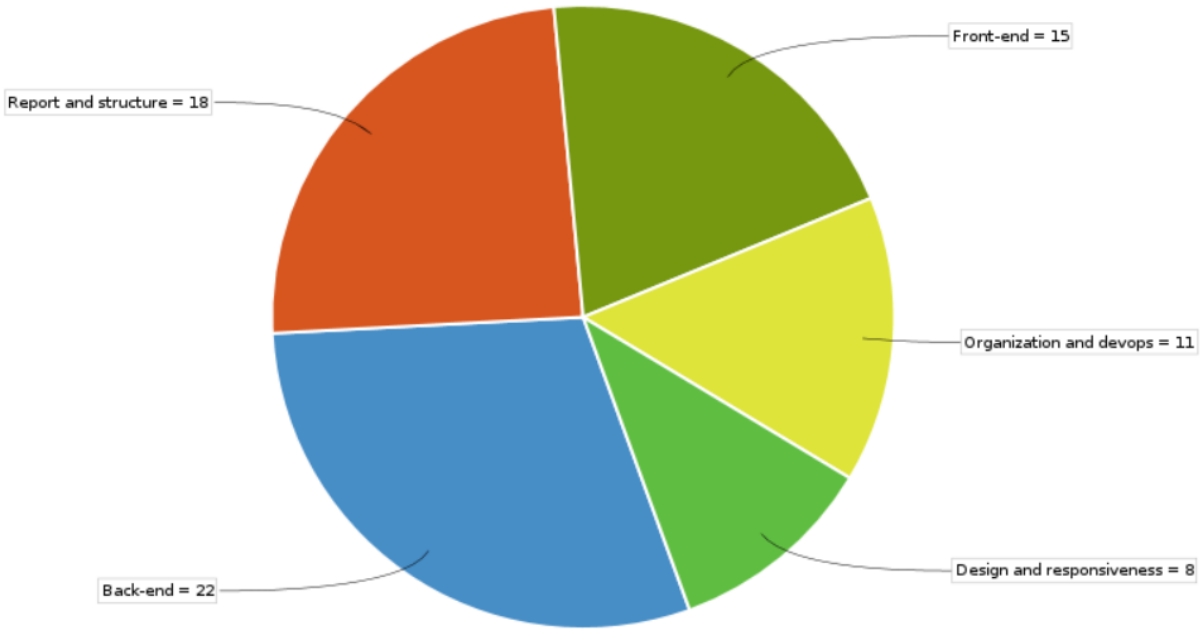


Figure 6.5: Components - Pie Chart.

Data Table

	Issues	%
Back-end	22	31%
Report and structure	18	25%
Front-end	15	21%
Organization and devops	11	15%
Design and responsiveness	8	11%

Figure 6.6: Components - Table.

Component	Description	Component lead	Default assignee	Issues
Report and structure	Report and structure of project	saidanur.sideif-zada	Component lead	18 issues
Organization and devops	Project management and devops	Ibrahimkhanli Murad	Component lead	11 issues
Front-end	Front-end of project	Farid Imanzade	Component lead	15 issues
Design and responsiveness	Design and responsiveness of project	Ofelya Rahmanova	Component lead	8 issues
Back-end	Logic behind front	Ibrahimkhanli Murad	Component lead	22 issues

Figure 6.7: Components - Data.

Pie Chart Report

Project: Se Project (Priority)

Chart

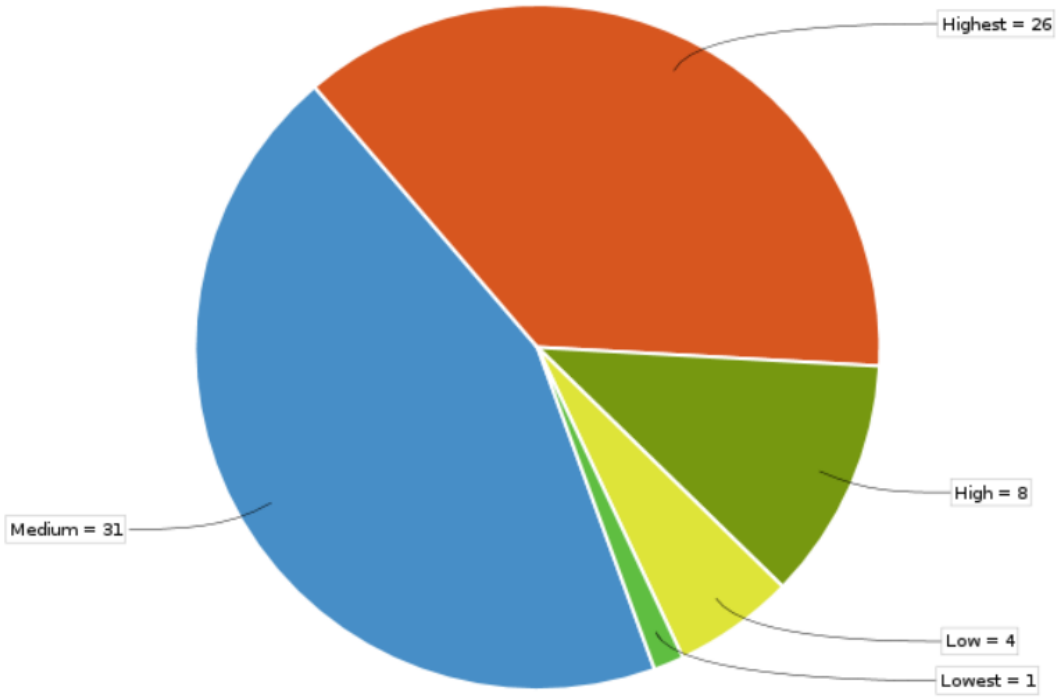


Figure 6.8: Priorities - Pie Chart.

Data Table

	Issues	%
Medium	31	44%
Highest	26	37%
High	8	11%
Low	4	5%
Lowest	1	1%

Figure 6.9: Priorities - Table.

Pie Chart Report

Project: Se Project (Labels)

Chart

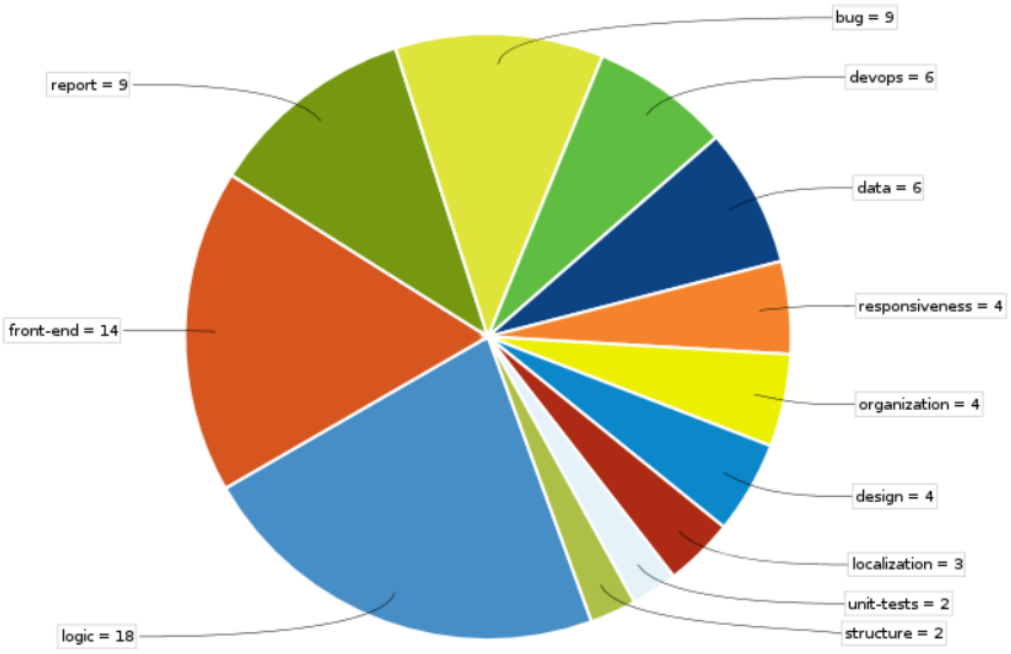


Figure 6.10: Labels - Pie Chart.

Data Table

	Issues	%
logic	18	25%
front-end	14	20%
report	9	12%
bug	9	12%
devops	6	8%
data	6	8%
responsiveness	4	5%
organization	4	5%
design	4	5%
localization	3	4%
unit-tests	2	2%
structure	2	2%

Figure 6.11: Labels - Table.

Pie Chart Report

Project: Se Project (Sprint)

Chart

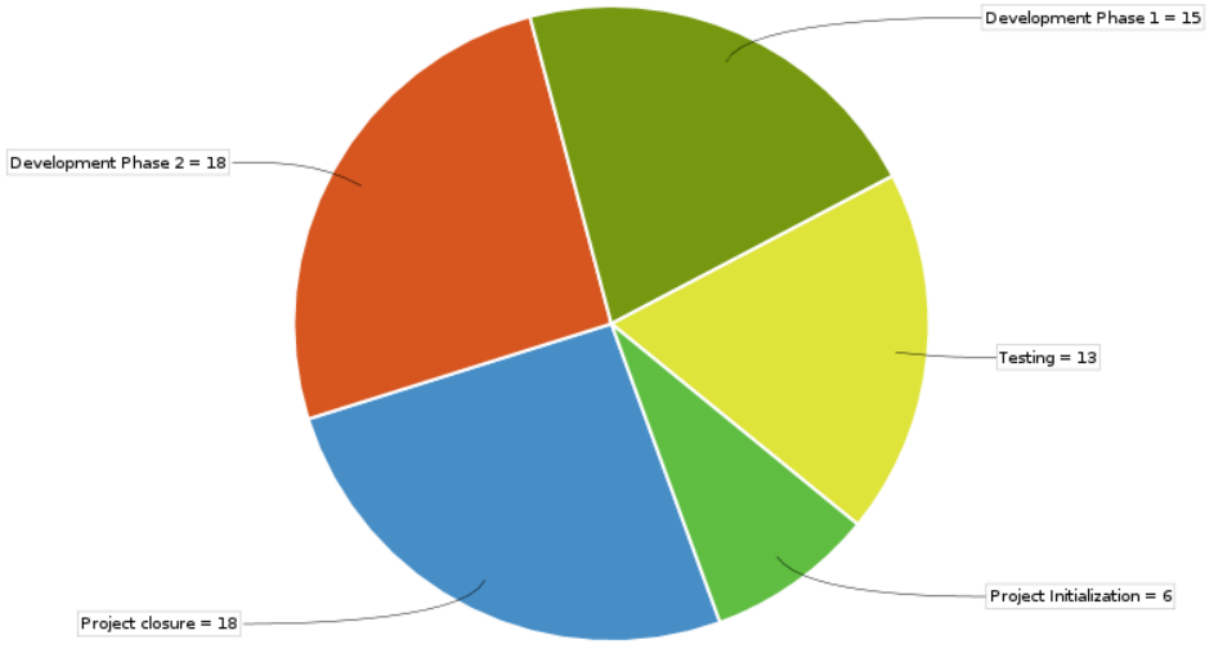


Figure 6.12: Sprints - Pie Chart.

Data Table

	Issues	%
Project closure	18	25%
Development Phase 2	18	25%
Development Phase 1	15	21%
Testing	13	18%
Project Initialization	6	8%

Figure 6.13: Sprints - Table.

Conclusions

What we learnt?

Nowadays most of the things has it is online equivalent, and one of the popular products in virtual world is board games which easily replaces face-to-face games. The board game industry has seen a resurgence in recent years and is expected to carry on growing further. Thus, working on "Love Letter" game gave us valuable knowledge and experience in a various areas.

Throughout the procedure of the project development, formation of web applications has been profusely practiced, hence sharpening our coding skills. Withal, we had an opportunity to have first-hand knowledge of the beauty of working on real-life project, as well as facing the difficulties of efficient cooperation. We predominantly worked on the core principals of the team development, which allowed us to easily communicate, as well as respect each other's opinion while making significant decisions that has huge impact on project's success. The main aspects we worked on were bringing the whole team to one vision, making team to work as one, making everyone to have the same end goal and to help each other to reach it.

We investigated the characteristics of a good software system, and considered what a development process would need to include to build such software.

Being a small team of 4 members, choosing "Agile" as a Software Development Life Cycle model to deliver qualitative project on time using resources we have, seemed appropriate to us, as it has a lot of advantages which facilitated our work. This method allowed us to learn a lot of new things, to adapt to conditions, to focus on essentials during each iteration. Despite the fact that while setting up the development process we've met some difficulties, the chosen model was efficient for us.

What we did?

As a result of our project we did:

- Practiced and improved ourselves on various areas of software development
- Gained skills in developing effective software
- Practiced with Agile methods
- Developed effective teamwork skills
- Proposed an implementation of Love Letter - card game
- Demonstrated as much as possible the qualities of a good Man-Machine Interface
- Localized interface for 4 languages
- Formed a coherent "whole" User Interface
- Implemented Artificial Intelligence module for game
- Made application responsive for mobile
- Implemented continues integration and deployment
- Implemented unit tests
- Wrote detailed and quality report
- Proposed user manual and graphs

Result

The developed "Love Letter" game can be accessed by visiting:
https://loveletterapp.herokuapp.com/