**LogiWord**

**Brain training Android Application**

**Bachelor’s degree in software engineering**

**Project report**

**Group:**

**Akos Faddi – 253992**

**David Kabaly – 253785**

**Krzysztof Majcher – 253784**

**Supervisor:**

**Kasper Knop Rasmussen**



**Software Engineering**

**Bachelor**

**20-12-2019**

**Table of contents**

[Appendix A – Use case descriptions 1](#_Toc27490219)

[Appendix B – Client Class Diagram 5](#_Toc27490220)

[Appendix C – Server Class Diagram 6](#_Toc27490221)

[Appendix D – Package Diagram 7](#_Toc27490222)

[Appendix E – Sequence Diagram 8](#_Toc27490223)

[Appendix F – Architecture Diagram 9](#_Toc27490224)

[Appendix G – Predesign Figma plans 10](#_Toc27490225)

[Appendix H – Scrum 11](#_Toc27490226)

# Appendix A – Use case descriptions

|  |  |
| --- | --- |
| Use Case | Multiplayer Match |
| Actor | User |
| Description | The user will compete against an opponent to complete a word and achieve more points than the enemy. |
| Precondition | The user logged in and is online |
| Postcondition | The user played a multiplayer match and gets a score |

|  |  |
| --- | --- |
| Use Case | Challenge a friend |
| Actor | User |
| Description | The user can select an opponent from the friend list to play a match against each other. They will get a word and should make the most points out of the word based on mathematical calculations. |
| Precondition | The user logged in and is online |
| Postcondition | The user played a multiplayer match against a friend |

|  |  |
| --- | --- |
| Use Case | Random Match |
| Actor | User |
| Description | The user will get a random opponent to play a match against. They have to earn the most points with limited mathematical signatures. |
| Precondition | The user logged in and is online |
| Postcondition | The user played a multiplayer match against a random opponent |

|  |  |
| --- | --- |
| Use Case | Classic Mode |
| Actor | User |
| Description | The user will get limited resources like few numbers and operations and tries to get as many points as possible |
| Postcondition | The user played a single-player match and gets a score |

|  |  |
| --- | --- |
| Use Case | Daily Challenge |
| Actor | User |
| Description | The user will get a word daily that needs to be completed with the given resources |
| Precondition | The user logged in and is online |
| Postcondition | The user played a daily challenge and gets a score |

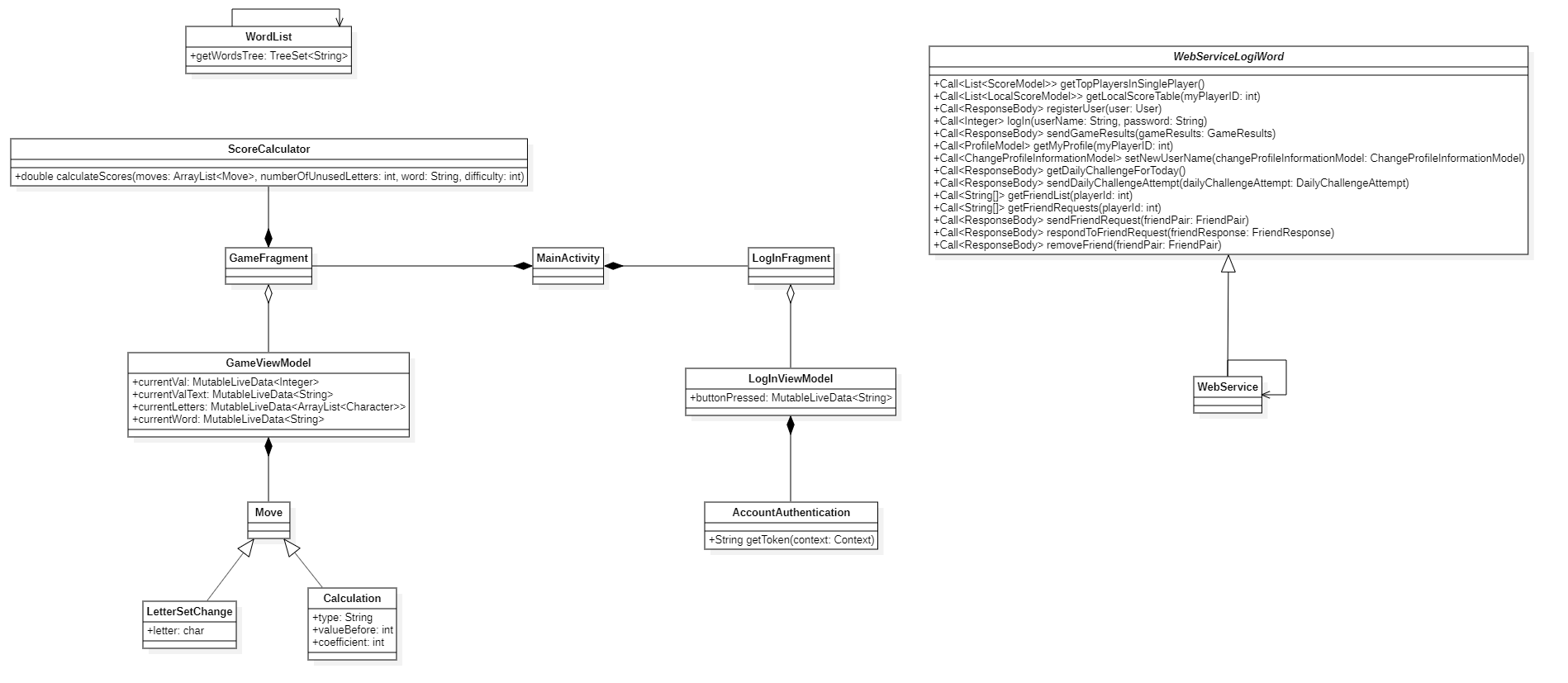
|  |  |
| --- | --- |
| Use Case | Multiplayer Scores |
| Actor | User |
| Description | The user can see a ranking list based on the scores in multiplayer. |
| Precondition | The user is online and logged in |
| Postcondition | The ranking list is displayed |

|  |  |
| --- | --- |
| Use Case | Single Player Scores |
| Actor | User |
| Description | The user can check the single-player scores what have been achieved. |
| Postcondition | The single-player scores are displayed |

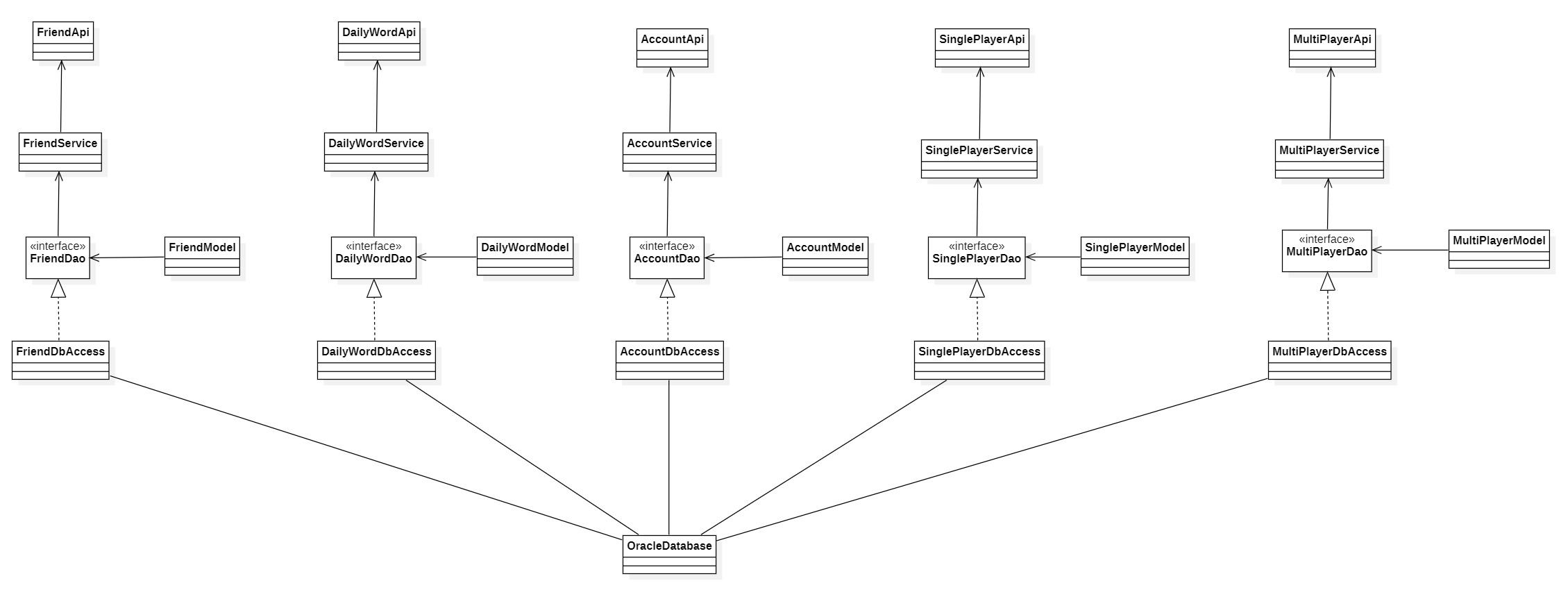
|  |  |
| --- | --- |
| Use Case | Tutorial |
| Actor | User |
| Description | User can access the prepared form of a tutorial that explains the basics mechanics of the application. This option will be highlighted if the user is using the system for the first time. |
| Postcondition | The user played the tutorial match |

|  |  |
| --- | --- |
| Use Case | Friend List |
| Actor | User |
| Description | User can access the friend list where all other players will be displayed what the user added. The online or offline status also will be shown in there and the invites from other users. It is possible to add a friend based on the given email address in the system. In this menu, the user can also remove a friend. |
| Precondition | The user is online and logged in |
| Postcondition | The user managed the friend list |

# Appendix B – Client Class Diagram

****

# Appendix C – Server Class Diagram

****

# Appendix D – Package Diagram

**A screenshot of a cell phone

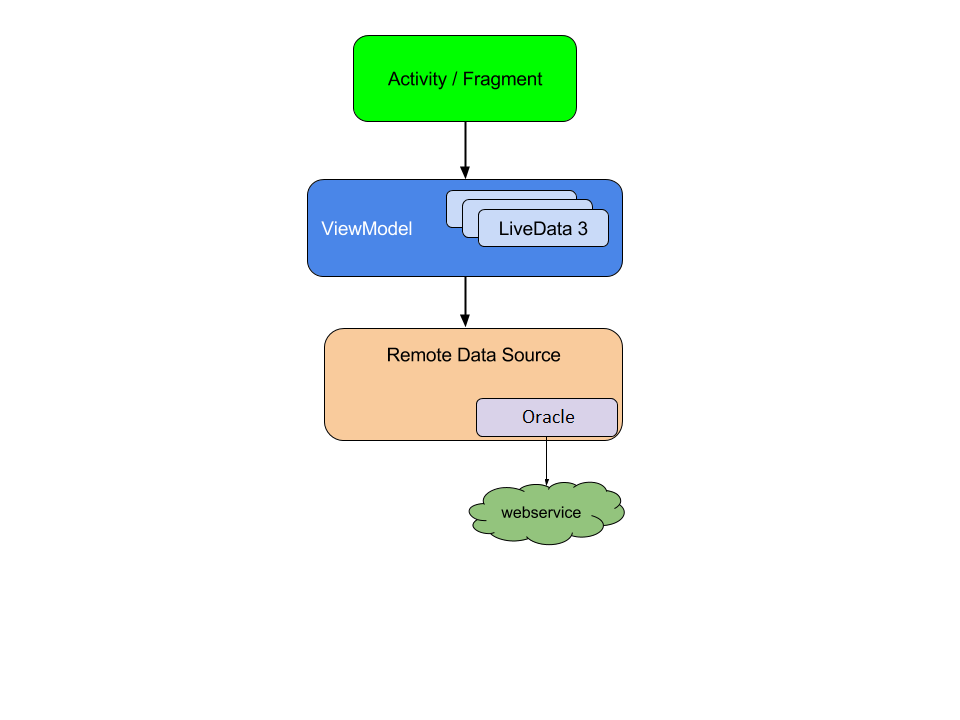
Description automatically generated**

# Appendix E – Sequence Diagram

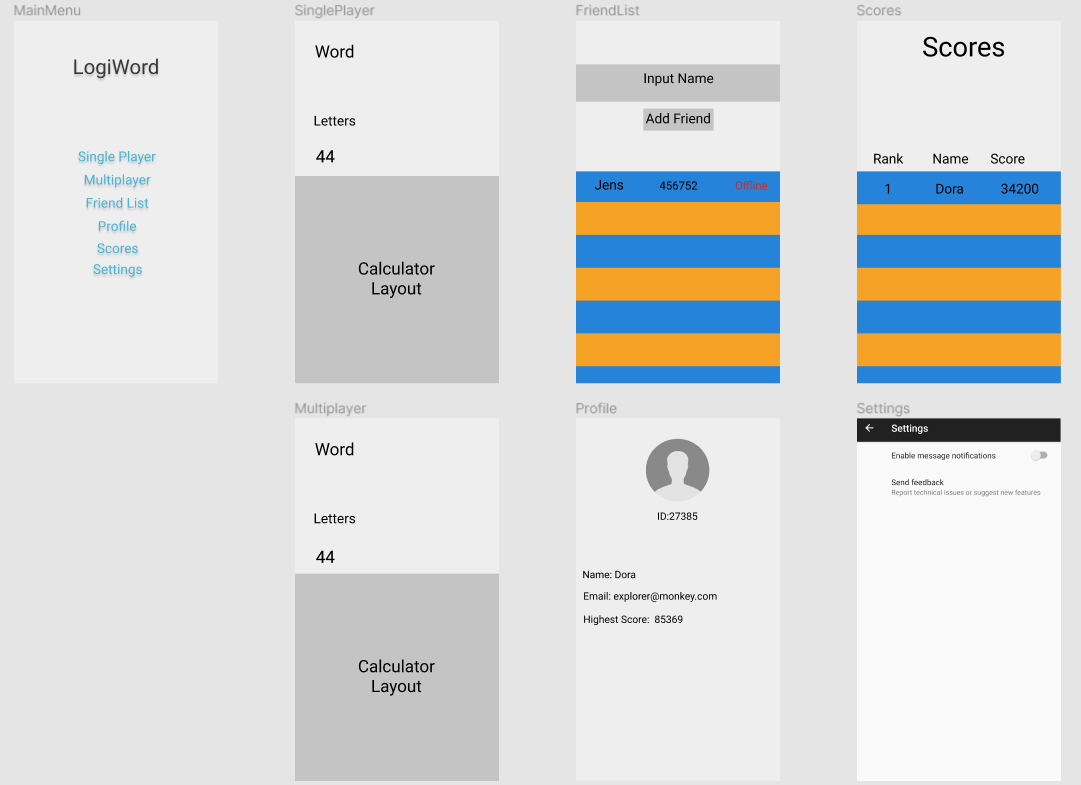
**A screenshot of text

Description automatically generated**

# Appendix F – Architecture Diagram

****

# Appendix G – Predesign Figma plans

****

# Appendix H – Scrum

**Product backlog**

The product backlog contains the tasks that we set as our goals for this project. The story points represent the expected hours of work to complete the tasks.

|  |  |  |
| --- | --- | --- |
| Product backlog | Story points | Status |
| ER diagram and description | 7 | Done |
| Package diagram | 3 | Done |
| Architecture diagram | 1 | Done |
| Class diagrams | 6 | Done |
| Interaction diagrams | 5 | Done |
| UI design | 4 | Done |
| UI design description | 3 | Done |
| Identify design patterns and describe them | 2 | Done |
| Introduction | 2 | Done |
| MosCow | 5 | Done |
| Requirement description | 1 | Done |
| Domain model and entities | 8 | Done |
| Use case documentation | 8 | Done |
| Background description | 2 | Done |
| Implementation documentation | 20 | Done |
| Test documentation | 15 | Done |
| Appendix | 10 | Done |
| Project report closing chapters | 15 | Done |
| Process report | 25 | Done |
| Package diagram fix | 0.5 | Done |
| Architecture diagram fix | 0.5 | Done |
| Description of technologies | 3 | Done |
| Client-side class diagram fix | 3 | Done |
| Server-side class diagram fix | 6 | Done |
| Change UI design | 2 | Done |
| Rich picture | 5 | Done |
| Acronyms and abbreviations | 3 | Done |
| Dimensional modelling | 5 | Done |
| MVC and MVVM | 12 | Done |
| Update use case documentation | 4 | Done |
| Update pattern documentation (dao, adapter) | 7 | Done |
| Update ER description | 1 | Done |
| Change rich picture | 3 | Done |
| Solid principle | 3 | Done |
| Analysis intro | 1 | Done |
| UI design documentation | 3 | Done |
| Server technologies | 2 | Done |
| Revision of project report | 65 | Done |
| References, table of contents, etc | 10 | Done |
| Small chapters | 5 | Done |
| System features | 3 | Done |
| Database implementation | 5 | Done |
| Menu implementation | 5 | Done |
| Basic server setup | 30 | Done |
| Connect client and server | 15 | Done |
| Game UI implementation | 5 | Done |
| Core game mechanics implementation | 55 | Done |
| Single Player Api | 15 | Done |
| Single Player Client | 10 | Done |
| Daily word API | 15 | Done |
| Daily word client | 5 | Done |
| Scores server | 20 | Done |
| Scores client | 20 | Done |
| Account management client | 15 | Done |
| Account management Server | 15 | Done |
| FriendList client | 5 | Done |
| FriendList Server | 8 | Done |
| Multiplayer client | 30 | Not implemented |
| Multiplayer server | 30 | Not implemented |
| Challenge friend client | 10 | Not implemented |
| Challenge friend server | 5 | Not implemented |
| Cache the data on the client | 30 | Not implemented |
| Tutorial | 20 | Not implemented |
| Achievements Client | 20 | Not implemented |
| Achievements Server | 8 | Not implemented |
| Hints | 10 | Not implemented |
| Multilanguage client | 10 | Not implemented |
| Multilanguage Server | 10 | Not implemented |
| History tab client | 20 | Not implemented |
| History tab server | 10 | Not implemented |
| Animations | 5 | Done |
| Testing | 20 | Done |
| Research Spring, JPA and DAO | 10 | Done |
| Fixing Toolbar for fragments | 3 | Done |
| Reviewing codes | 7 | Done |
| Setting menu | 5 | Done |
| Loading screen | 5 | Done |
| Friend list with recycle view | 5 | Done |
| Retrofit research | 5 | Done |
| Basic retrofit mvvm implantation | 25 | Done |
| DAO and dependency injection | 8 | Done |
| Research spring security | 5 | Done |
| Score recycle view implementation | 5 | Done |
| Api to update profile information | 3 | Done |
| Api to get daily word | 3 | Done |

**Sprints**

Each sprint we selected some of the tasks from the product backlog and added them to the current sprint’s sprint backlog. Most of the sprints were 1 week long.

**Sprint 1: 20.10.2019 – 22.10.2019**

**Planning:** This sprint is shorter than the other ones, because we agreed that each Wednesday will mark the start of a new sprint, as this is the day when we get feedback from our supervisor. The goal during this sprint is to have some of the diagrams that give a better understanding for the overall system.

|  |  |  |
| --- | --- | --- |
| Sprint backlog | Story points | Status |
| ER diagram and description | 7 | Done |
| Package diagram | 3 | Done |
| Architecture diagram | 1 | Done |

**Sprint review:** The objectives for this were achieved.

**Sprint retrospective:** We worked well together, but it was a short sprint and there were not many tasks.

**Sprint 2: 23.10.2019 – 29.10.2019**

**Planning:** For this sprint, we have to fix some of diagrams that were considered done in the previous sprint. The rest of the sprint will focus on the diagrams that could help with the architecture of the system.

|  |  |  |
| --- | --- | --- |
| Sprint backlog | Story points | Status |
| UI design | 4 | Done |
| Package diagram fix | 0.5 | Done |
| ER diagram description | 1 | Done |
| Architecture diagram fix | 0.5 | Done |
| Client-side class diagram | 3 | Done |
| Server-side class diagram | 3 | Done |
| Design patterns description | 2 | Not Implemented |
| Description of technologies | 3 | Not Implemented |
| Sequence diagrams | 5 | Not Implemented |

**Sprint review:** The fixes and majority of the diagrams were completed, the remaining tasks are close to finish, but they will have to be moved to the next sprint.

**Sprint retrospective:** We should dedicate more hours to the bachelor

**Sprint 3: 30.10.2019 – 05.11.2019**

**Planning:** In the previous sprint we finished most of the diagrams, so we are ready to take on some tasks that are about implementation. At first, we need the core of the app to be done and later we can extend it with everything else.

|  |  |  |
| --- | --- | --- |
| Sprint backlog | Story points | Status |
| Client-side class diagram fix | 3 | Done |
| Server-side class diagram fix | 6 | Not implemented |
| Design patterns description | 2 | Not implemented |
| Description of technologies | 3 | Not implemented |
| Sequence diagrams | 5 | Not implemented |
| Database implementation | 5 | Done |
| Change UI design | 2 | Not implemented |
| Menu implementation | 5 | Not implemented |
| Resources to letters | 25 | Not implemented |
| Letters to word | 20 | Not implemented |
| Giving points to the user | 10 | Not implemented |
| Setup basic API on server | 25 | Not implemented |
| Client connection to the server | 15 | Not implemented |
| Save data to the DB | 5 | Not implemented |

**Sprint review:** A most of the implementation is going nicely, but they cannot be considered done.

**Sprint retrospective:** Too many tasks have been included in this sprint

**Sprint 4: 06.11.2019 – 12.11.2019**

**Planning:** In the previous sprint we included way too many tasks, so most of those tasks are simply just moved to this sprint.

|  |  |  |
| --- | --- | --- |
| Sprint backlog | Story points | Status |
| Server-side class diagram fix | 3 | Not implemented |
| Design patterns description | 2 | Implemented |
| Description of technologies | 3 | Implemented |
| Sequence diagrams | 5 | Not implemented |
| Change UI design | 2 | Not implemented |
| Menu implementation | 5 | Implemented |
| Numbers to letters | 25 | Implemented |
| Letters to word | 20 | Not implemented |
| Giving points to the user | 10 | Not implemented |
| Client connection to the server | 15 | Not implemented |
| Setup basic API on server | 25 | Implemented |
| Save data to the DB | 5 | Not implemented |
| Game UI | 5 | Implemented |

**Sprint review:** The implementation is going well, and we managed to squeeze in some documentation too.

**Sprint retrospective:** Judging sprint points needs to be improved, but distributing the tasks is going great.

**Sprint 5:** **13.11.2019 – 19.11.2019**

**Planning:** Some of the implementation was missing in the previous sprint, those are added to this sprint and a few new tasks are added as well in case the previous tasks are finished, or someone needs something to do.

|  |  |  |
| --- | --- | --- |
| Sprint backlog | Story points | Status |
| Save data to the DB | 5 | Not implemented |
| Research Spring, JPA and DAO | 10 | Done |
| Api for single player save and retrieve | 15 | Not implemented |
| Sequence diagrams | 5 | Done |
| Implement animation | 5 | Done |
| Fixing Toolbar for fragments | 3 | Done |
| Reviewing codes | 7 | Done |
| Letters to word | 20 | Done |
| Word Validation | 5 | Done |
| Single Player Client | 10 | Not implemented |
| Friend list with recycle view | 5 | Not implemented |

**Sprint review:** The core mechanics are finally done, documentation is going well, now the focus will need to be on connecting the different system parts.

**Sprint retrospective:** Judging the sprint points will need to be improved further.

**Sprint 6: 20.11.2019 – 26.11.2019**

**Planning:** The focus needs to be on connecting the system elements, but a few documentation elements have been added, in case we have some extra time.

|  |  |  |
| --- | --- | --- |
| Sprint backlog | Story points | Status |
| Save data to the DB | 5 | Done |
| Api for single player save and retrieve | 15 | Done |
| Setting menu | 5 | Done |
| Loading screen | 5 | Done |
| Friend list with recycle view | 5 | Done |
| Retrofit research | 5 | Done |
| Basic retrofit mvvm implantation | 25 | Not implemented |
| Project introduction | 2 | Done |
| Rich picture | 5 | Not implemented |
| MoSCoW prioritization | 5 | Done |
| Requirement description | 1 | Done |
| Use case Pre description | 1 | Done |
| Single Player Client | 10 | Done |
| MVC and MVVM | 12 | Done |

**Sprint review:** A lot of things can be considered done, the ones that are not finished yet are going to be moved to the next sprint.

**Sprint retrospective:** The tasks should be distributed a little bit better. There were members who got a lot of tasks regarding documentation and not much about implementation.

**Sprint 7: 27.11.2019 – 03.12.2019**

**Planning:** Implementing new features and we should improve the already existing code quality. Some documentation was added as well.

|  |  |  |
| --- | --- | --- |
| Sprint backlog | Story points | Status |
| Research spring security | 5 | Done |
| Api for login and registration | 15 | Done |
| Project Report - Domain entities | 4 | Done |
| Project Report - Domain model | 4 | Done |
| Score recycle view implementation | 5 | Done |
| Basic retrofit mvvm implantation | 25 | Not implemented |
| Rich picture | 5 | Done |
| Acronyms and abbreviations | 3 | Done |
| Choice of technologies description | 1 | Done |
| Dimensional modelling | 5 | Done |
| DAO and dependency injection | 8 | Done |

**Sprint review:** Almost every task has been completed, very satisfied with the results.

**Sprint retrospective:** Only a few more sprints left, we should include more documentation tasks.

**Sprint 8: 04.12.2019 – 10.12.2019**

**Planning:** This sprint is going to be the last sprint where we are writing any code, because the next one should focus on documentation only, so every feature that we want in the system should be implemented now.

|  |  |  |
| --- | --- | --- |
| Sprint backlog | Story points | Status |
| Update use case documentation | 4 | Not Implemented |
| Basic retrofit mvvm implantation | 25 | Done |
| Retrofit for score management | 20 | Done |
| Scores API | 20 | Done |
| Retrofit for account management | 20 | Done |
| Account management API | 15 | Done |
| Update pattern documentation (dao, adapter) | 7 | Not Implemented |
| Api to update profile information | 3 | Done |
| Api to get daily word | 3 | Done |
| Api for friend list | 8 | Done |
| Profile for client | 15 | Done |
| Daily challenge for client | 7 | Done |
| Friend list for client | 7 | Done |
| Update ER description | 1 | Not Implemented |
| Background description | 2 | Done |
| Change rich picture | 3 | Done |
| Solid principle | 3 | Done |
| Analysis intro | 1 | Done |
| UI design documentation | 3 | Not Implemented |
| Server technologies | 2 | Not Implemented |

**Sprint review:** All code related thing that had to be implemented is done, some extra documentation is done as well.

**Sprint retrospective:** The team did well, lot of tasks have been completed.

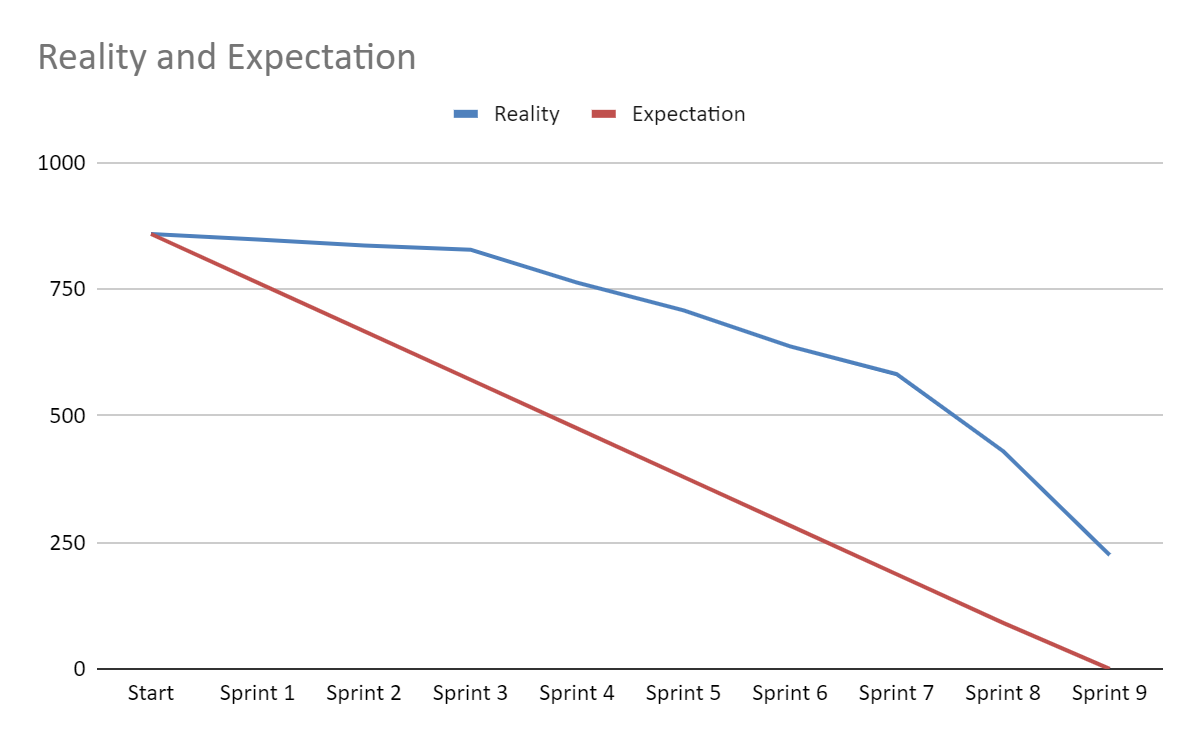
**Sprint 9: 11.12.2019 – 20.12.2019**

**Planning:** This is the last sprint, so it should focus entirely on the documentation. It is a little bit longer than the other sprints, because the remaining time wouldn’t be enough for a normal sprint.

|  |  |  |
| --- | --- | --- |
| Sprint backlog | Story points | Status |
| Update use case documentation | 4 | Done |
| Update pattern documentation (dao, adapter) | 7 | Done |
| Update ER description | 1 | Done |
| UI design documentation | 3 | Done |
| Server technologies | 2 | Done |
| Implementation documentation | 20 | Done |
| Test documentation | 15 | Done |
| Revision of project report | 65 | Done |
| References, table of contents, etc | 10 | Done |
| Appendix | 10 | Done |
| Project report closing chapters | 15 | Done |
| Process report | 25 | Done |
| System features | 3 | Done |
| Small chapters | 5 | Done |

**Sprint review:** Everything is done, ready to hand in.

**Burndown chart**

****