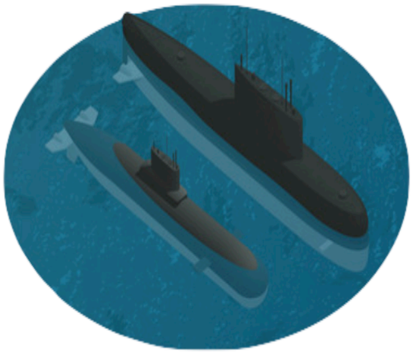
University of Illinois Chicago

Software Engineering I

CS 440 / Fall 2020

Score: 89/100
Insufficient Content. There is not list of figures/tables.
Diagrams are not labeled. Document needs additional formatting.



BATTLE LINES:
THE WAR OF SUBMARINES

Prepared by Group 17 :

Viktor Kirillov, Mykyta Parovyi, Volodymyr Vakhniuk, Aleksandra Dmitrieva

I. Project Description

II.Requirements

III.Design

IV. Test Plans

V. Project Issues

I. Project Description

1. Project Overview

Battle Lines: The War of Submarines is an extension to a popular, classic game of Battleship. The main goal of an extension is to create an environment where the game can be played by more than two people at the same time. This extension provides game rules and mechanics for two - four players, giving them new and unique board layout, ability to move their ships and a new goal: destroy opponents bases instead of ships in order to defeat another player

2. The Purpose of the Project

a. The User Business or Background of the Project Effort

Current implementation of the *Battleship* game provides a game mechanics suited for two players only. That makes it impossible to play in a group of three or four players without a significant amount of changes in both game logic and board layout.

b. Goals of the Project

We want to make it possible to play a classic *Battleship* game with more than two players at a time.

c. Measurement

The project is considered as completed when it provides an ability for two, three and four users to play a fair game, considering that the game lasts for at least 10 minutes.

3. The Scope of the Work

The work that is covered by the project encapsulates the activities associated with the user's entertainment while playing the *Battleship* game with more than two players.

a. The Current Situation

The game industry provides a lot of games that can be played with more than two players at a time, but if the user wants to play a classic or an old game (*Battleship* in our case) with more than one of the friends, they currently have no ability at all to do that.

b. The Context of the Work

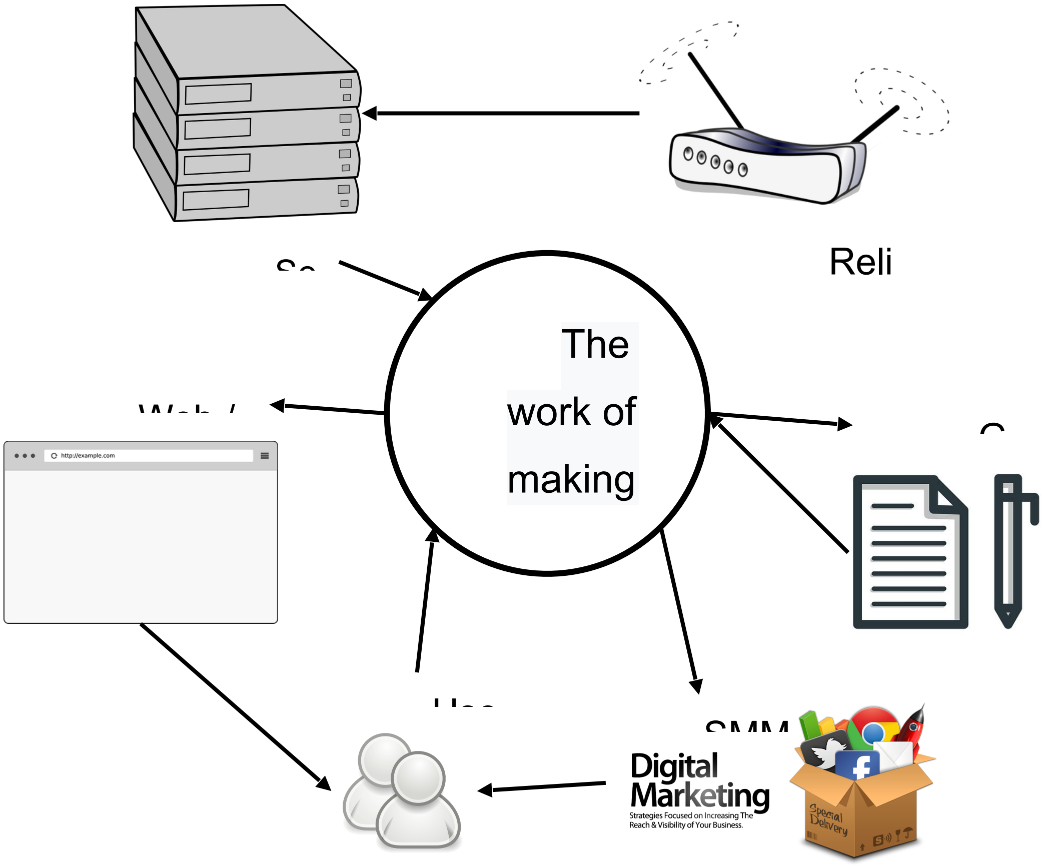

Reli
The
work of
making
http://example.com
Digital
Marketing
Strategies Focused on Increasing The Reach & Visibility of Your Business.
Special Delivery

Figure 1 - The context of the work

Besides creating game rules, the scope of the work also includes providing the interface for the user to be able to connect and interact with. In order to make it possible for multiple players to interact with each other within the game, the reliable server should be created and configured to handle user connections and send the game data between all players.

c. Work Partitioning

Business Event List:

Event Name	Input & Output	Summary
Server warning or error	Server (in) Web / mobile interface (out)	Warn users about the risk of losing the game progress and start the maintenance.
User connects to the game	User connection (in) Web / mobile interface (out)	Display the game interface to the user.
Lack of users	SMM & DM (out)	Target potential users with a promotion on social media or game-related web pages.
User reports a problem or suggests a feature	Game rules (out) Game rules (in)	Adjust the game rules and implement them in the game.

d. Competing Products

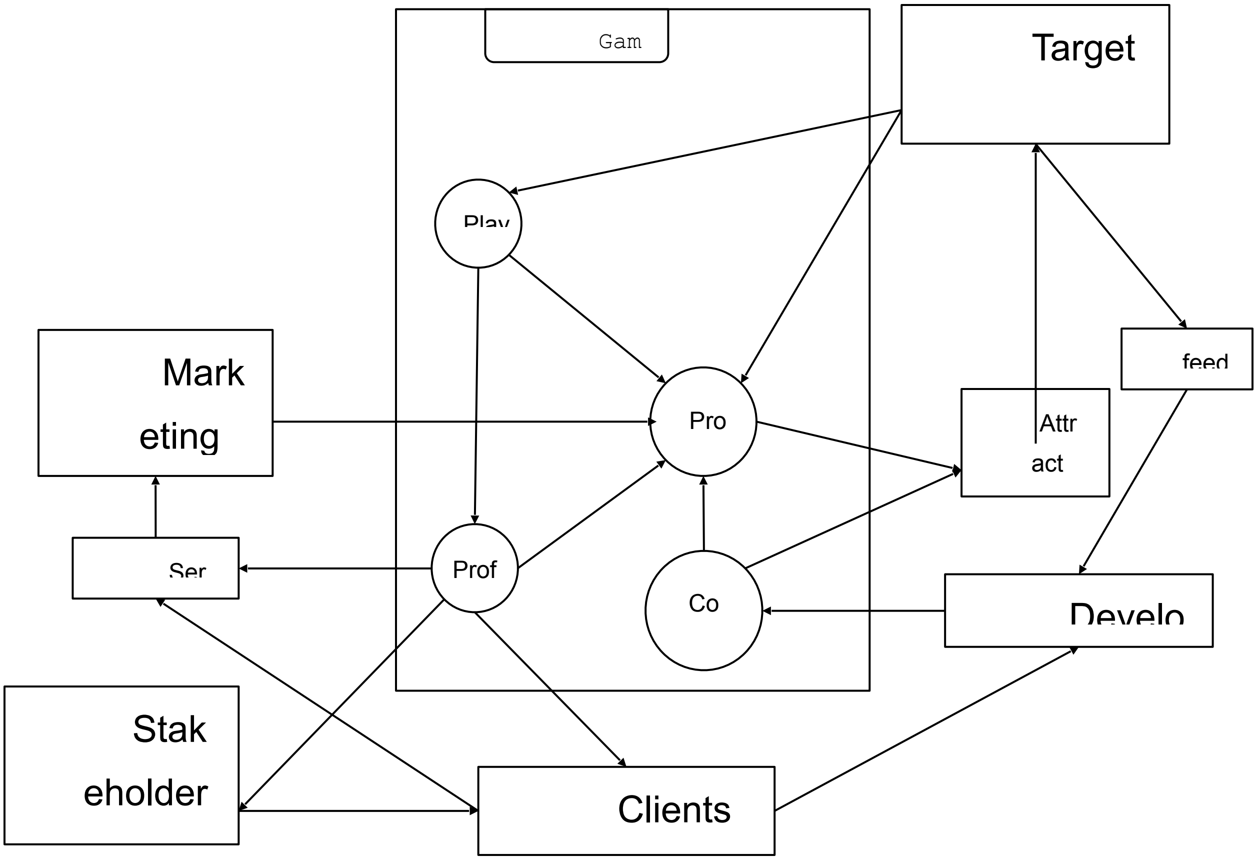
No products that give an ability to play classic or close to classic Battleship games with more than two players were found.

4. The Scope of the Product

Subset of work:

- Client Playing
- Promotion via Marketing Services
- Stakeholders Involvement
- Developers work

a. Scenario Diagram(s)



b. Product Scenario List

1. Game Playing - **Target Audience**
2. Promotion & Target Audience Attraction
3. **Marketing Services / Stakeholders / Developers / Target Audience / Client**
4. Content Updating - **Developers / Target Audience / Stakeholders**
5. Profit Getting - **Stakeholders / Clients**
6. Feedback Creation - **Target Audience**

a. Individual Product Scenarios

✳***Game Playing:*** A Target Audience person who purchased the product is able to use it at any time by playing it. First thing they have to do is to log into their account and select to either play in a single or multiplayer mode. Secondly depending on what they chose, the game starts and they either begin their battle with a computer (if a single player mode was chosen) or with another enemy player (if a multiplayer mode was chosen). When the game ends, players can either play again or quit the game.

✴***Promotion & Target Audience Attraction:*** Target Audience, Stakeholders, Developers & Marketing services are all involved. Target Audience promote the product by using it and occasionally telling other people about it.
Clients pay the Marketing services to do their job of promoting it. |
Marketing services promote the product by using social media ads, google ads, YouTube ads, etc. Developers promote the product by updating its content from Target Audience feedback of the product.Stakeholders promote the product via their website and social media platform announcements of funding.

✴***Content Updating***: Developers get the feedback from Target Audience and update content accordingly.

✴***Profit getting:*** Clients and Stakeholders will get their profit from Target Audience purchasing the product.

✴***Feedback Creation:*** Target Audience will generate feedback while using the product which will directly be addressed to developers to look at.

5. Stakeholders

There is an extremely large pool of groups that are interested in game development. Below listed all the groups that are considered the most important.

a. The Client

Battle Lines: The War of Submarines is the original idea of VMVA Development Group Organization.

VMVA Development Group takes full interest in development and realizing the game.

The game is intended for common use. Main purpose of the project is entertainment.

b. The Customer

The game is intended to be placed on online platforms such as Kongregate.com, MmoGames.com for free and demo version. The game also will be available for purchase. Customers purchasing the game are hand-on users.

c. Hands-On Users of the Product

Hands-On Users of the product can be divided into several groups:

Gamers:

Gamers are the people who play the game. The concept of Battle Line: war of submarines is a new concept in the war games category. Individual gamers are responsible for the game set up and get themself familiar with additional rules the game might have. Since the Battle Line: war of submarines is an expansion of classic battleship games it is assumed that many gamers will have some knowledge of the rules. For every gamer who is unsure about the rules we provide a description of the game rules. The game is created for people of all ages but the target audience is children 7 yo and up.

Game Market:

The Battle Line: war of submarines is a new game to be released in the online games market. Therefore, the Battle Line: war of submarines game is to be a market research subject.

Parents:

The Battle Line: war of submarines is suitable for young age players and parents are often concerned on how the game is influencing their child behavior and education.

The Battle Line: war of submarines is a strategy type game. The hands on user is the one who is interested in intellectual entertaining games.

Gender:

The game is suitable for any gender and educational level user who has an access to the computer with ethernet connection.

d. Maintenance Users and Service Technicians

The VMVA Development Group takes full responsibility of the game as well as technical support, game maintenance and updates development.

e. Other Stakeholders

The Battle Line: war of submarines game product is focusing on entertaining. However, the game requires users to develop strategy and therefore can carry educational purpose. Teachers and other educators may use the game as a smart entertaining experience for their students. Game elements, problem-solving skills, and discussions with fellow team members aid in facilitating active learning amongst end-users in the global market.

f. User Participation

The Battle Line: war of submarines is a complex development game. There are several stages in the development process prior final release.

Stages include: requirements, design, implementation, verification, maintenance. User participation is desirable at every stage of the process.

Several demo versions are to be developed and released for hand on users to use and give feedback on. The demo games will differ in design implementation as well as difficulty of the rules. Besides the user feedback VMVA Development Group is interested in performance evaluation of the game. Based on uses r evaluation the version decision is to be made.

g. Priorities Assigned to Users

VMVA Development Group is a non-profit educational organization. However the development and maintenance process requires time and resources. VMVA Development Group priorities customers purchasing the game from online platforms and hand-on users purchasing full game for personal use.

6. Mandated Constraints

a. Solution Constraints

Description: The product should be a web-based game and function correctly in Google Chrome and Mozilla Firefox browsers.

Rationale: Users are likely to have different browsers and operating systems.

Fit criterion: The product shall be accessible from a browser.

Description: The product should properly handle the disconnection of any of the players and shouldn't crash for other players.

Rationale: Users are likely to have internet issues or the product may have unexpected code flow.

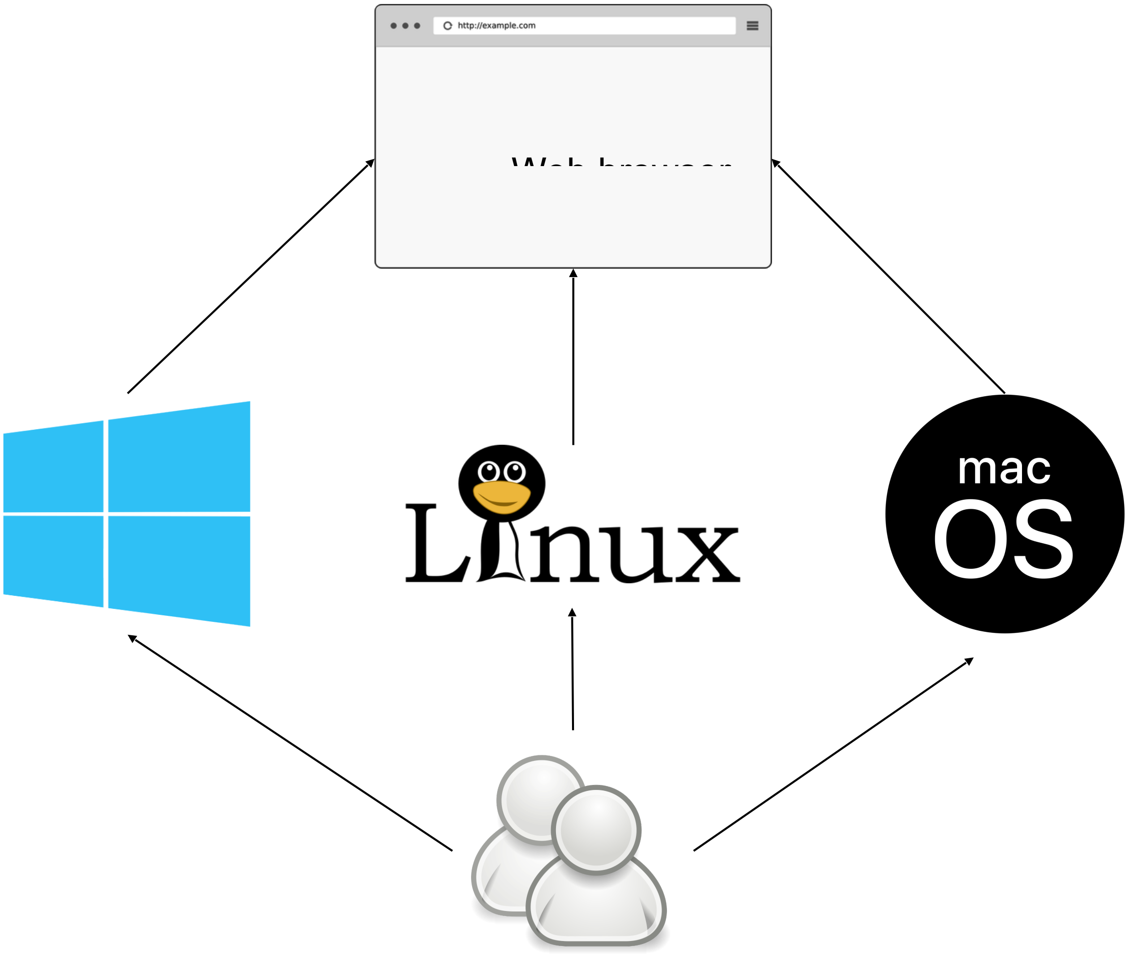
Fit criterion: The product shall continue operating correctly and count disconnected players as a defeated one.

Description: The product should operate correctly under a big number of users using it at a time.

Rationale: The product tends to be unique, therefore a big amount of users are expected.

Fit criterion: The product shall operate properly under 100,000 connected users.

b. Implementation Environment of the Current System



Users should be able to access the product no matter what operating system they use. The product should be able to function properly on Windows, MacOS and Linux operating systems. It is assumed that users have a good internet connection.

c. Partner or Collaborative Applications

The product should have an ability to create an account and/or login using both Google and Discord authentication services. The product is not intended to have any of the data that should be used outside of itself.

d. Off-the-Shelf Software

The product has no requirements for any of the COTS software to be included. It is up to developers of the product to decide which software to use in the development and release phases.

e. Anticipated Workplace Environment

This product should be used in any environment regardless of surrounding conditions.

f. Schedule Constraints

There are no essential deadlines for this project.

g. Budget Constraints

Expected budget is 20,000$

7. Naming Conventions and Definitions

a. Definitions of Key Terms

This section is intended to clarify and define all terms used in the Battlerine: war of submarines game.

Base: base of operations for a naval fleet. Illustrated as buildings

Submarines: a watercraft capable of independent operation underwater.

Fleet: consist of submarines and belongs to the base

International waters: term used to describe territory that does not belong to any player and used to position a fleet.

b. UML and Other Notation Used in This Document

This document uses Graphical Notation Reference to build diagrams.

The Unified Modeling Language™ (UML®) is a standard visual modeling language intended to be used for:

- modeling business and similar processes,
- analysis, design, and implementation of software-based systems

UML is a common language for business analysts, software architects and developers used to describe, specify, design, and document existing or new business processes, structure and behavior of artifacts of software systems.

Specifications Explained that process:

- provides guidance as to the order of a team's activities,
- specifies what artifacts should be developed,
- directs the tasks of individual developers and the team as a whole, and
- offers criteria for monitoring and measuring a project's products and activities.

Given some UML diagram, we can't be sure to understand the depicted part or behavior of the system from the diagram alone. Some information could be intentionally omitted from the diagram, some information represented on the diagram could have different interpretations, and some concepts of UML have no graphical notation at all, so there is no way to depict those on diagrams.

For example, semantics of multiplicity of actors and multiplicity of use cases on use case diagrams is not defined precisely in the UML specification and could mean either concurrent or successive usage of use cases.

Name of an abstract classifier is shown in italics while final classifier has no specific graphical notation, so there is no way to determine whether classifier is final or not from the diagram

8. Relevant Facts and Assumptions

a. Facts

According to "Strategy Games Market Research" conducted in 2018 (NEW YORK, Dec. 11, 2018 /PRNewswire) report on global board games market offers analysis on market size & forecast, market share, industry trends, growth drivers, and vendor analysis. The exponential growth of the entertainment and gaming industry is attributing to the rising demand for analog games in the market. According to the research strategy games influence factors such as problem-solving skills and critical thinking in team-oriented games allow players to build communication and relationship skills with consumers in the market. The global board games market is anticipated to reach values of more than $12 billion by 2023, growing at a CAGR of over 9% during 2017-2023.

b. Assumptions

The development of the Battle Line: war of submarines is based on strong assumption of the growing strategy games market. The popularity of classic battleship games give us a strong belief that adding new features and expanding into multiplayer concepts will bring attention not only from hands-on users(gamers) but also will attract online game markets to research and invest into the development of the Battle Line: war of submarines. Educational features of the game can not be underestimated. The game requires problem-solving skills and critical thinking as well as team-oriented skills and communication skills. Taking this fact into account it is assumed that games can be used for educational purposes. The Battle Line: war of submarines game will be created to serve entertaining purpose therefore it is to be associated with leisure and good time. The game design supports users' experience providing exceptional visual and sounds effects. Detailed design gives the user full experience. The Battle Line: war of submarines is a war-themed board game for multiplayer therefore there is a large number of players who are going to play the game at one time. It is assumed that every user will have ethernet access and suitable technology such as smartphones, tablets or laptops.

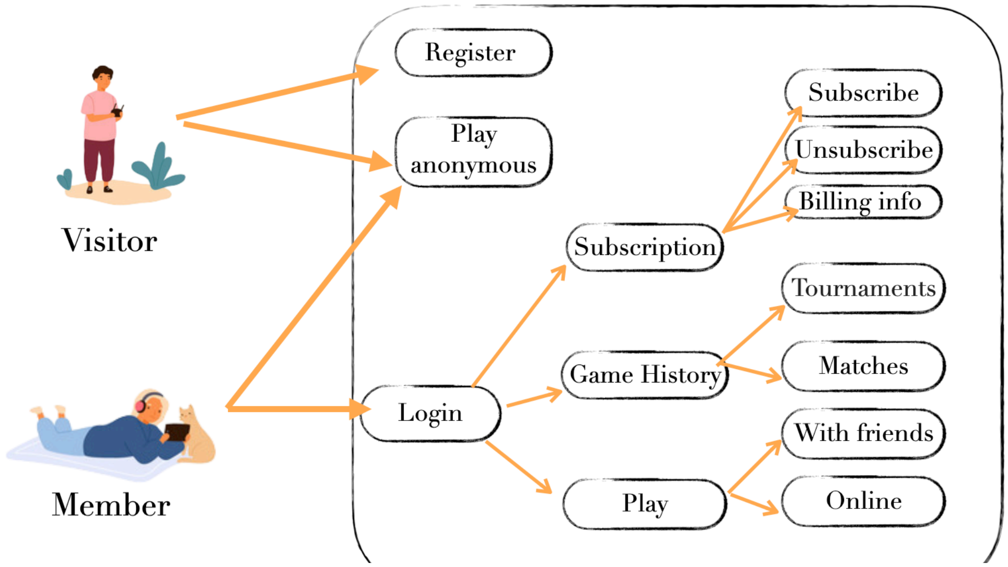
The game can be a platform for commercial advertising.

II. Requirements

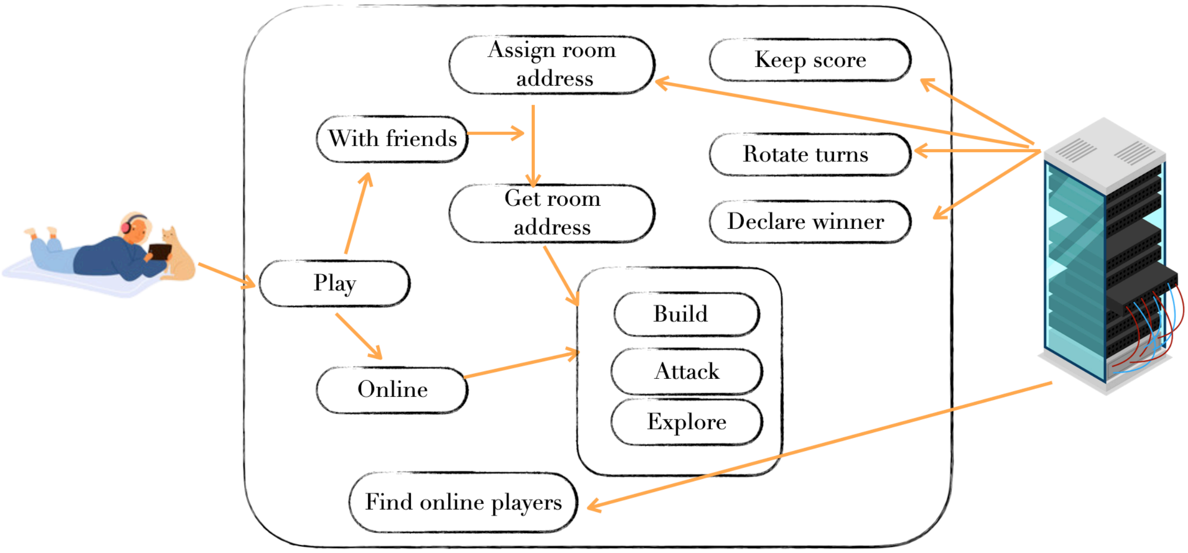
1. Product Use Cases

a. Use Case Diagrams

PERSONAL ACCOUNT



GAME PROCESS



b. Product Use Case List

- User account manipulations:
  ✓Register

    - ✓Login
    - ✓Charge user

- Start New Game
    - Two Players OR MultiPlyer
        - ✓Request Random Player
        - ✓Connect With Friend
        - ✓Play Against AI

- In Game
    - ✓Build new Game Items
    - ✓Attack the opponent

- DB
    - ✓Store new Data
    - ✓Request Data
    - ✓Update Data
    - ✓Return Data

- Game System
    - ✓Generate New Game Field
    - ✓Keep Score
    - ✓Initialize players connection

c. Individual Product Use Cases

Use Case #2	RequestRandomPlayer
Description	Players request to start a game with randomly picked player available on the server
Actors	Player1, Player2, server, game system
Assumptions	The game request and player connection is successful. Enters new scene .GameField implements .TwoPlayerGame
Steps	user request game by pressing “Two player game” button
Flow of Events	.twoPl waits for user to press .twoPl sends server request to initialize connection to opponent .gameState updated to two player game state server initialize connection between players
Exit conditions	Game was successfully loaded and system exited .PickTheMode scene

Use Case #3	ConnectWithFriend
Description	Players request to start a game with a friend
Actors	Player1, server, game system
Assumptions	The game request and player connection with friends is successful. Enters new scene .GameField implements .TwoPlayerGame
Steps	user request game by pressing “Friend game” button
Flow of Events	.friendG waits for user to press .friendG sends server request to initialize connection .gameState updated to friends game state random code generated and displayed on screen. when code entered by player’s friend connection initialized
Exit conditions	Game was successfully loaded and system exited .PickTheMode scene

Use Case #4	InGame
Description	Game was successfully initialized
Actors	Player1, Player2, (or more up to 4 when extends .MultiPlayer) server, game system
Assumptions	Game system successful keeps score while sending game state to the server
Enter Conditions	successful game field initialization
Flow of Events	server initialize a connection, game logic starts
Exit conditions	.GameState.currentState changes to GAME_OVER flag

Use Case #5	BuildNewItem
Description	Players request to place new object on the .GameField
Actors	Player1, .GameField scene

Assumptions	The object was successfully placed on the .GameField .GameState is updated and server recorded changes
Flow of Events	.twoPl waits for user to press .twoPl sends server request to initialize connection .gameState updated to two player game state server initialize connection between players
Exit conditions	New item appears on the field

Use Case #6	Attack
Description	Players request to make an Attack
Actors	Player1,Player2, server, game system
Assumptions	The attack was made and successfully analyzed. the result of the attack was recorded and displayed to all players
Steps	player 1 picks location to attack and initialize attack by pressing “Attack” btn
Flow of Events	attack was initialize, implemented and result was recorded and displayed
Exit conditions	Attack was implemented Attack finished results displayed

Use Case #7	Charge Customer
Description	This use case charges the order currently being placed to a credit card.
Actors	User, server, game system, bank System
Assumptions	The purchase is successfully made, or user is notified of frailer.

Steps	1. The use case begins when a user selects "Credit Card" as a payment option, while in use case Place Order 2. The system requests the credit card number, type and expiration date. 3. The user enters the information. 4. The system verifies that the credit card is valid for the amount to be charged and completes the credit card transaction. The system stores the payment details and returns a success message
Errors to handle	If the credit card cannot be validated the use case ends, returning a failure message
Exit conditions	The customer has been charged for the order.

2. Functional Requirements

- **Database**: Server shall have full database functionality for data manipulation and storage. That functionality must include: *Requesting, Setting, Updating and Returning* data.

➡ Fit Criterions:

  - If a specific data was requested, then the db should only return that data, not anything else.
  - All queries should be processed appropriately, giving the right data as the result

- **Server**: Server shall be able to handle two or more users and all interactions between them (shooting, moving the ships, placing new ones, etc.) as well as keeping a steady socket connection with each client in the game in order to pass the data back and forth instantly, without losing any client connection.

➡ Fit criterions:

  - Passing updated client data to the server should not cause any errors, but reflect them on all other clients by sending updated data through the server to them.

  - Connecting two or more players to a game should not cause any errors, but successfully assign them to an instance of game and keep a steady socket connection.
  - If there are 2 players waiting for the game to be found, the server should connect them without any errors and start the game session.

- **Authentication**: Server should have an Authentication system that stores personal game information of each user as well as their login credentials they filled during the registration form.

➡ Fit criterions:

  - Requests for a specific user data should only return that user’s data not any other one.
  - Setting or Updating a particular user data entry in the database should not cause any other users’ updates or sets apart from that particular user entry.
  - Users should be able to login with their credentials without any errors, apart from those that include mistyping and other similar ones.

- **Client**:
  - Client shall display the right map to a user.
  - Client shall generate new game fields during each new game the user starts.
  - Client shall handle all server incoming data and correctly reflect that updated data on the game field during each game.
  - Client shall keep a score of each player during the game.

➡ Fit criterions:

  - Client should not reveal the enemy map to a user at any time during the game.
  - Client should not generate any errors while processing server incoming data from the other players in an on-going game

3. Data Requirements

a. System should store

i. Encrypted User Credentials

➡ Rationale:

Privacy and security of the personal data.

This also allows the system to have login/register functionality.

➡ Fit Criterion:

Original data incoming from the user should be ensured that it should not be visible in its raw (decrypted format) anywhere in the system. User data should strictly be encrypted in the whole system.

ii. Game Entities Data

➡ Rationale:

This allows us to avoid getting modified data from the client on the server which would mean cheating. Thus, keeping every entity specific data on the server such as (damage of a submarine dealt, displayed part of the map, max number of available units on the map, etc.) keeps us safe and away from data modifications.

➡ Fit criterion:

Make sure there is no crucial data coming from the client to the server that can be modified in order to get some sort of an advantage.

iii. Game Sessions

➡ Rationale:

In order for a system to maintain properly a game process between players it should store its game session along with players entities coordinates, status of bases and submarines on the map.

➡ Fit criterion:

In case if a player disconnects and connects back due to the temporal internet connection loss, a player should be placed back to the game process with the same state they had before they got disconnected.

4. Performance Requirements

a. Speed and Latency Requirements

The purpose of this section of the document is to outline the Software Performance Goals for Product Battle Lines: The War of Submarines. These are the goals that are based on general advice on response time from ***Jason Nielsen book on Usability.*** Where the 10 seconds limit for response is considered maximum to keep user focused/uninterrupted.

Any interface between a user and the automated system shall have a maximum response time of 3 seconds.

- Server needs to be available at any time
- Connection between random players needs to be initialized within 10 seconds.
- Connection with friend limit is 1.5-3 second
- The response shall be fast enough to avoid interrupting the user's flow of thought.
- The database request should be fulfilled within 1.5 seconds

The system should limit all wait times to 10 seconds, which is about the limit for keeping the user's attention focused on the dialogue. For longer delays, users will want to perform other tasks while waiting for the computer to finish, so they should be given feedback indicating when the computer expects to be done. Feedback during the delay is especially important if the response time is likely to be highly variable, since users will then not know what to expect.

Considerations: Response time may vary based on the ethernet connection on the user's end.

b. Precision or Accuracy Requirements

This section of report is dedicated to accuracy of game system performance.

- It is expected that the game would perform the logic correctly.

- AI systems must have an ability to generate game strategy based on the players level of performance determined by experience and games previously won.
- Server is assumed to be provide strong and uninterrupted connection between the players

c. Capacity Requirements

This section specifies the volumes that the product must be able to deal with and the amount of data stored by the product. Our game logic is designed to be stored on cloud and be dynamically loaded as the game progresses and develops into the local game directory. The game is not graphically heavy, therefore does not require large memory capacity.

To ensure that ethernet connectivity is required for games to be initialized.

HARD DRIVE: At least 5 GB of free space with at least 500 MB additional space for custom content and saved games.

5. Dependability Requirements

a. Reliability Requirements

- This product provides 90% confidence that all the data and game state would be saved in case of unexpected exit or disconnection.
- In case of server failure the game creates the local copy of the game state, so it would be saved in the server when connection is restored.
- This product provides an intuitive game navigation system.

b. Availability Requirements

Battle Lines is a desktop game is available for everyone who meets the set up and connection requirements:

- The product shall be available for use 24 hours per day, 365 days per year.

- Ethernet connection is required.

c. Robustness or Fault-Tolerance Requirements

Battle Lines: The war of submarines relies on storing game data into the server to insure player's account availability from any device. Therefore, connection to the server is required.

- Disconnection from the server may be considered an abnormal environment.
- Therefore, a product must be able to provide all of its functionality after or during some unusual situation.
- The product shall continue to operate in local mode whenever it loses its link to the central server.
- While disconnected from the server game system must create a copy of all game states changes for each player.

6. Maintainability and Supportability Requirements

a. Maintenance Requirements

- Development Group reserves the right to collect all the data related to system failure and use this information in maintaining purposes.
- Users would be notified via in game notifications.
- If an update is required for optimal system performance, the game should not be able to exit pop-up unless the update is installed.

b. Supportability Requirements

User support is available via email that can be found in the game settings.

c. Adaptability Requirements

The Battle Lines: The war of submarines is expected to run on the systems meeting this requirement:

- Memory:2 GB
- Graphics Card:NVIDIA GeForce 8500 GT
- CPU:Intel Pentium 4 2.00GHz
- SIMPLE GAME File Size:200 MB
- OS:Windows 7 +

d. Scalability or Extensibility Requirements

- The system must be able to handle around 10,000–12,000 players, 4000–500 active simultaneously.
- Game is expected to gain wide popularity within the first 5 years of release.
- If a players' number is expanded over the server limit, usage of an additional server is recommended.

e. Longevity Requirements

- During the first year, while the new game concept is being adopted by users, it is expected to have increased user support requests.
- Battle Lines:War of Submarines is expected to operate without major updates and within next 3 years.
- One major update every 3 year is recommended.

7. Security Requirements

a. Access Requirements

VMVA Development Group reserves the right to access all the data related to the project as well as modify and update if necessary.

- No body has access to the users account I info except the user who own the account.

- Every user with account has ability to look and check previous scores and game results by query.
- Data base access is limited to the members of the development group

b. Integrity Requirements

The Batteries: war of submarines guaranties safety of personal and financial data. To ensure safety program must support two end encryptions.
Consideration: detected fraud attempts will result in immediate consent to check the personal data.

c. Privacy Requirements

- Login and password information stays encrypted for all parties involved.
- User is guaranteed to have personal account where he can look up all personal information he provided, make changes or remove it.
- The result of the game between multiple players must stay private and available for users involved in it.
- All user information stays private unless consent was given. (System must request consent)

d. Audit Requirements

To maintain the system safe and efficient semi-yearly audit is required. An audit is "a systematic, independent and documented process for obtaining audit evidence and evaluating it objectively to determine the extent to which the audit criteria are fulfilled".
Audit criteria:

- Safe user environment.
- Ban on swearing words
- Encryption of personal data
- Efficient and informative notifications and emails

Consideration: Audit is required. Qualified personnel are allowed to do the audit. All audit information must be reported in unified form to VMVA director of quality control.

e. Immunity Requirements

The Battle Lines: war of Submarines is a server-based game that is require ethernet connection. The system must be equipped with strong antivirus system. User are guaranteed to have cyber safe environment while playing the game.

8. Usability and Humanity Requirements

The Battle Lines: war of submarines guarantees to have intuitive user interface following the UX standards. It is a game that excels by taking a simple, well-trod formula and executing it with incredible style. Every new scene of the game must follow the design standards. (Can be found in design standards section)

a. Ease of Use Requirements

The Battle Lines: war of submarines must rely on intuitive based navigation. Which includes common UI patterns such as:

- Hamburger menu symbol (leaving context hidden until needed)
- Icons must reflect its meaning (Ex: magnifying glass symbol equals searching)
- Full screen navigation options for navigating between game styles and rules.

- Gesture based navigation is used for interacting while game is in a progress

b. Personalization and Internationalization Requirements

The Battle Lines: war of submarines is internationally available game and must support different languages and cultural differences if necessary. System is adjustable for users with disability. In addition, game provides theme choice for user

- The system support at least 5 languages
- User has ability to pick the language of his choice.
- The system must provide enlarged objects for users with vision disabilities.
- The system must provide sound volume control
- The system is implemented in at least 3 color styles and theme adjustment is available for user.

c. Learning Requirements

The Battle Lines: war of submarines system provides clear description of game rules for all users.

- The system is recommended for users of age of 6 and up.
- Updated game rules must be always available for user during the game for the reference if needed without interrupting the game flow.
- All users age of 6 and up must be able to learn the game rules within one hour.
- No educational background is necessary for the users.

Constrains: Users not familiar with no basic computer skills might find some difficulties in navigation and might need extra training. VMVA Development Group can provide expanded navigation and rules guide for such user via request.

d. Understandability and Politeness Requirements

The Battle Lines: war of submarines is internationally available game and must support different languages and cultural differences if necessary.

- The game system must use symbols and words that are only socially acceptable.
- The symbol usage across the system must be checked for international political and ethical correctness.
- The game system must not use “swear” words
- The game system must guarantee safe environment for user at any location of the word.

e. Accessibility Requirements

The game system does not need to require users to sign up with account in order to have full access to the game. However, users are able to keep track of all the games played by signing up with an account.

- User reserves the right to play without creating an account
- Only signed users are able to see games history and scores
- All in game purchases are available for users with confirmed accounts
- Only users with confirmed account can participate in game chats

Considerations: Users without an account can make one-time purchases, however, the purchase would be gone when cycle is ended.

f. User Documentation Requirements

The Battle Lines: war of submarines’ game system must provide an access for the user where he can find all the information related to the game rules and environment navigation.

- User is quarantined to have access to brief game rules at any point of the game.
- Accessing rules must not affect the game flow.
- Documentation must be written in friendly and clear language
- Documentation must provide information on symbols used in the game.

g. Training Requirements

The Battle Lines: war of submarines does not require additional training that is not specified in user documentation provided for every user who has access to the game. Constrains: If additional training is necessary: expanded documentation can be requested via email.

9. Look and Feel Requirements

a. Appearance Requirements

- The Battle Line: War of submarines is an educational multiplayer game and must be designed in warm colors.
- The color-scheme must be appealing for all genders.
- The game must not have bright/neon colors.
- The game must be visually pleasing and need to avoid sharp figures and lines

b. Style Requirements

The game design and style must follow all the design requirements provided in “Design” documentation.

- The game needs to give an impression of cartooned battlefield
- Warm color-scheme is used to give user positive feeling from the game navigation and the game itself.
- The game should avoid loud, disturbing sounds
- The game style must be appealing for all users of different age (6+)

10. Operational and Environmental Requirements

a. Expected Physical Environment

The game requires to have computer with mouse navigation and keyboard

* No specific requirements are necessary.

* User is responsible for physical safety while playing the game

b. Requirements for Interfacing with Adjacent Systems

- The products shall work on the last four releases of the five most popular browsers
- The product must be cross platform and available on Linux systems and Windows systems

c. Productization Requirements

- The product must be downloaded from the server and installed by the user locally on the device of his choice.

-

d. Release Requirements

- Each release shall not cause previous features to fail
- User should be notified of every new release.
- Installation of new release is done via the server.
- If update is necessary for optimal performance, it can be forced.

11. Cultural and Political Requirements

a. Cultural Requirements

- The product shall not be offensive to religious or ethnic groups.
- The product shall keep a record of public holidays for all countries in the European.
- The product must be available in different languages.

b. Political Requirements

<u>The Battle Line: war of submarines must stay apolitical.</u>

12. Legal Requirements

12.a. Compliance Requirements

- Personal Information (username/password) should not be known to anyone and could not be accessed anywhere in the system, only in encrypted form.
- Rationale: To avoid personal data thefts
- Fit criterion: Ensure that it is impossible to access personal data and get it in decrypted form from anywhere in the system.

12.b. Standards Requirements

- Product should comply with GDPR standards.
- Product should be developed using agile software development methodologies.
- Product should be developed using any version control software to avoid delays during development.
- Product should comply with ISO/IEC/IEEE 29119-1 standards.

III. Design

1. System Design

a. Design goals

Portability

- Server can be installed on a variety of machines and operating systems and functions in a variety of networking environments

Transparency

- The server might itself be distributed, but should provide a single "logical" service to the user

Scalability

- Server has spare capacity to handle larger number of clients

Flexibility

- Should be usable for a variety of user interfaces

Reliability

- System should survive communication link problems

Efficiency

- System must be efficient by reusing existing materials while building original work

2. Current Software Architecture

Client-server pattern

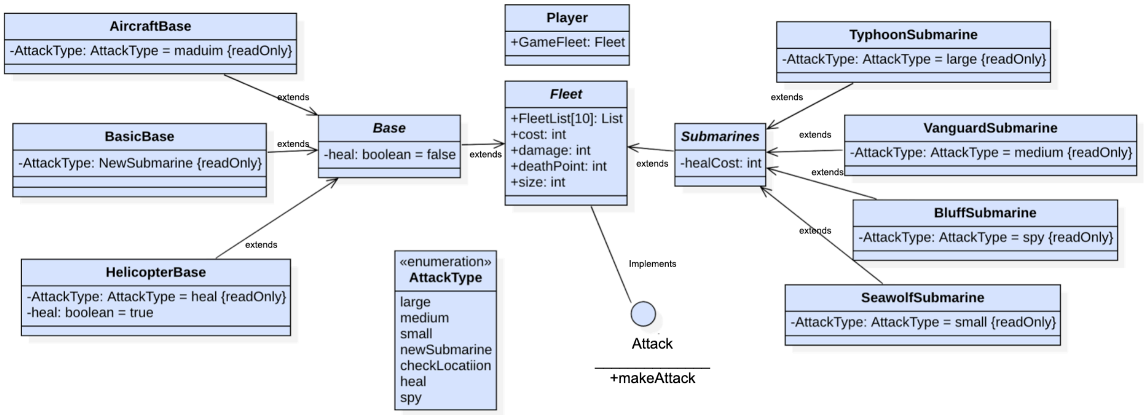
3. Proposed Software Architecture

a. Overview

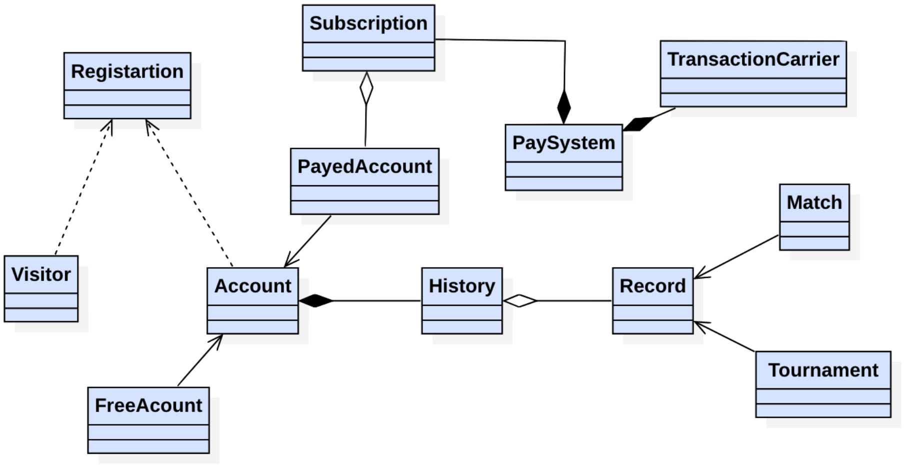
This pattern consists of two parties; a **server** and multiple **clients**. The server component will provide services to multiple client components. Clients request services from the server and the server provides relevant services to those clients.. .

b. Class Diagrams

IN GAME CLASSES



ACCOUNT ARCHITECTURE



SINGLE CLASS DIAGRAM EXAMPLE

BasicBase

-AttackType: AttackType = newSubmarines {readOnly}
-heal: boolean = false {readOnly}
-Size: int = 6 {readOnly}
-Cost: int = 40 {readOnly}
-health: int = 20

+makeAttack(AttackType)
+createSubmarine(Submarine: Submarine)
+takeDamage(int)

TyphoonSubmarine

-AttackType: AttackType = large {readOnly}
-heal: boolean = true {readOnly}
-Size: int = 4 {readOnly}
-Cost: int = 16 {readOnly}
-health: int = 8

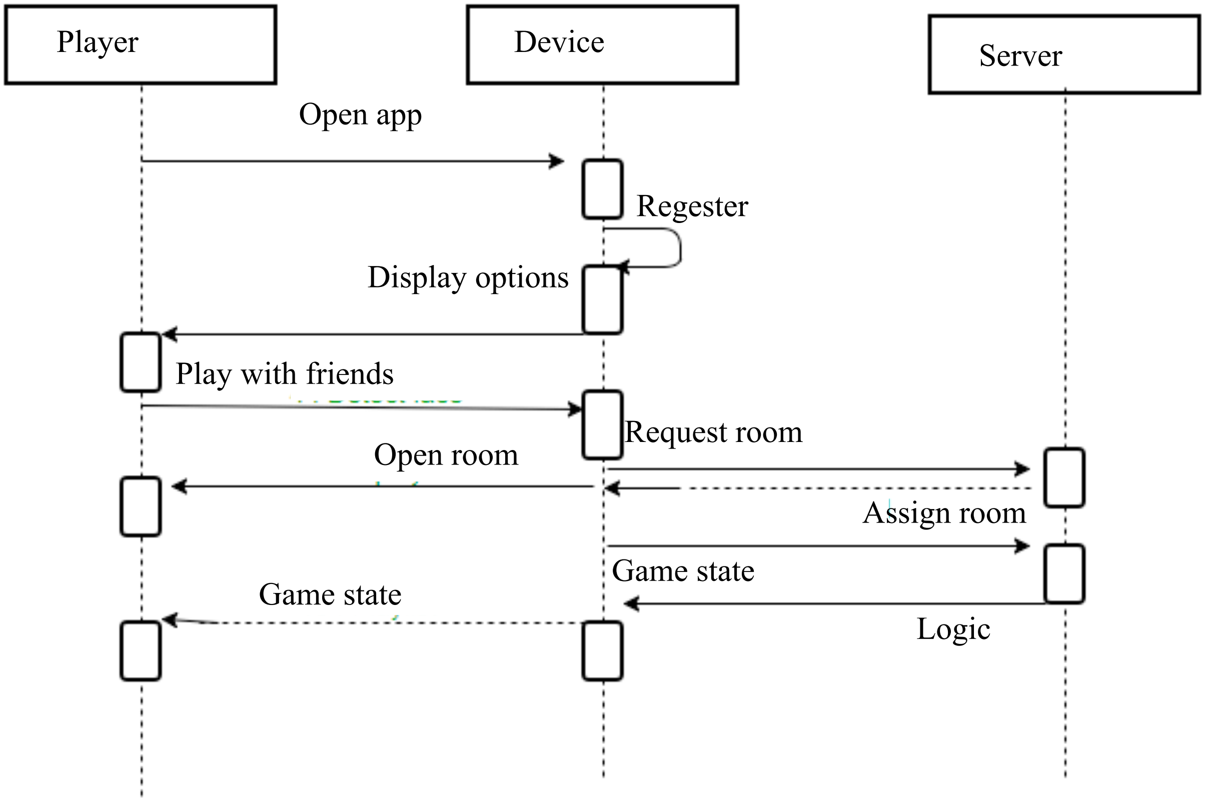
+makeAttack(AttackType)
+takeDamage(int)

Account

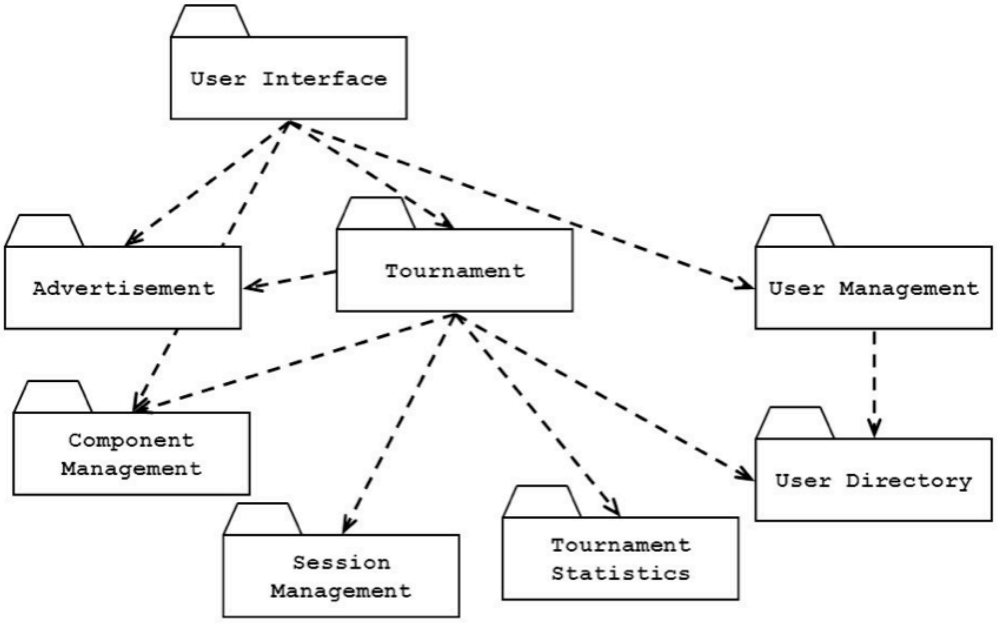
-Type: acountType
-History: List of records
-Subscription: bollean = false

+subscribe()
+unsubscribe()
+lookUpHistory()
+startNewGame(GameType: GameType)
+playAnonymous()

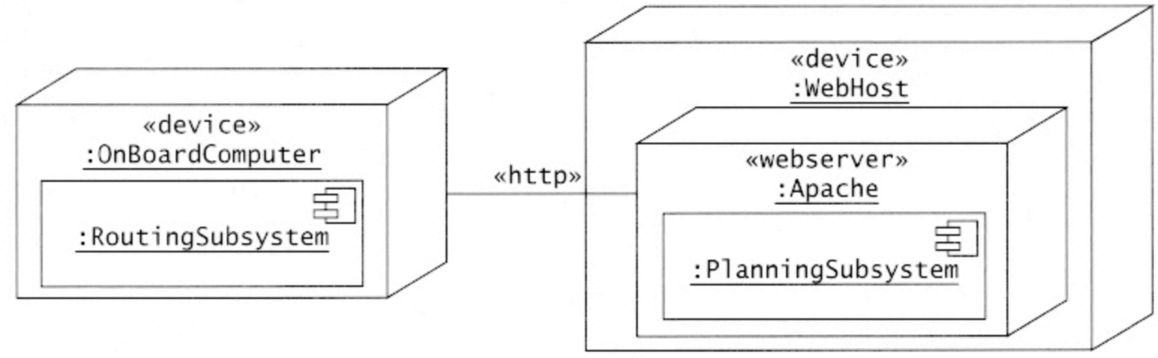
c. Dynamic Model



d. Subsystem Decomposition



e. Hardware / software mapping



f. Persistent Data management

Any data base storage such as SQL, MongoDB

g. Access control and security

User access accounts via unique user name and password.

User personal info is encrypted.

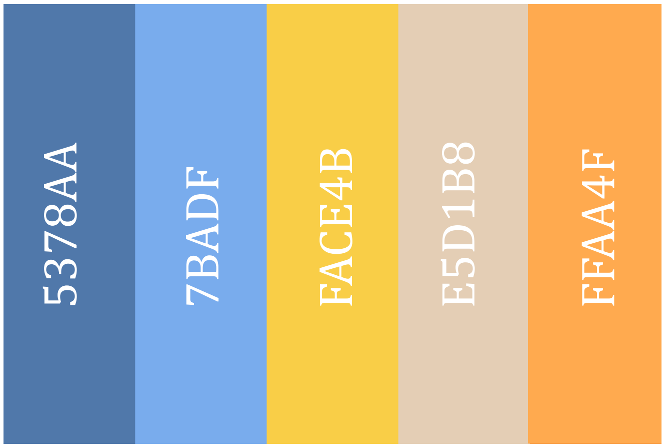
4. Subsystem services

User interface is easy and readable for user to efficiently communicate with device of his choice.

5. User Interface

Clean and minimalistic. Game must be appealing to user with different background and interests.

HEX UI colors:

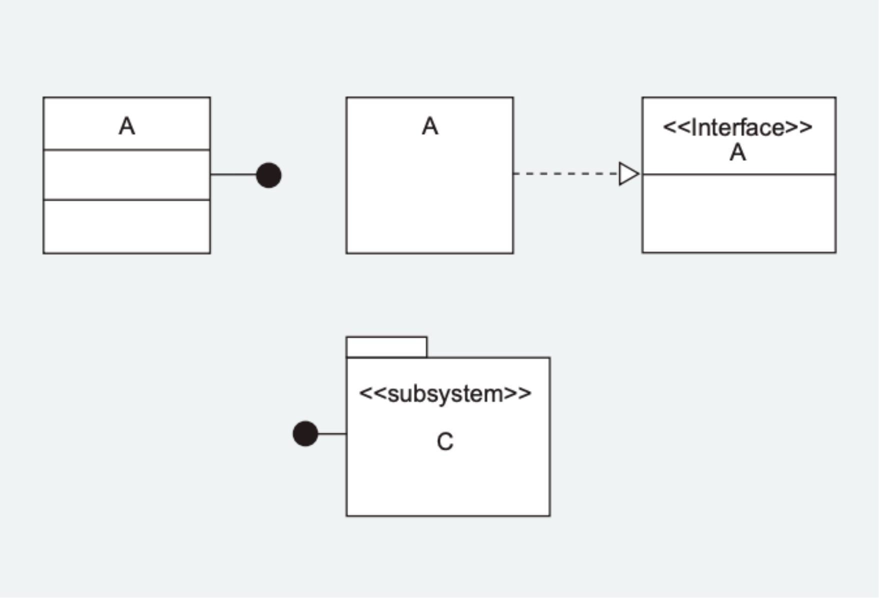


6. Object Design

a. Object Design trade-offs

Cost: with addition of more features and ML development cost growth.

b. Interface Documentation guidelines

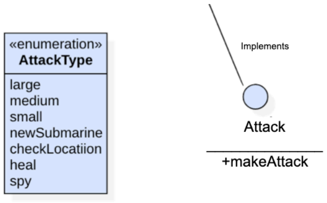


UML uses a “lollipop” to denote an interface, which can be appended to classes and subsystems among other things.

c. Packages

Game package
Server package
Logic Package
Connection

d. Class Interfaces



IV. Test Plans

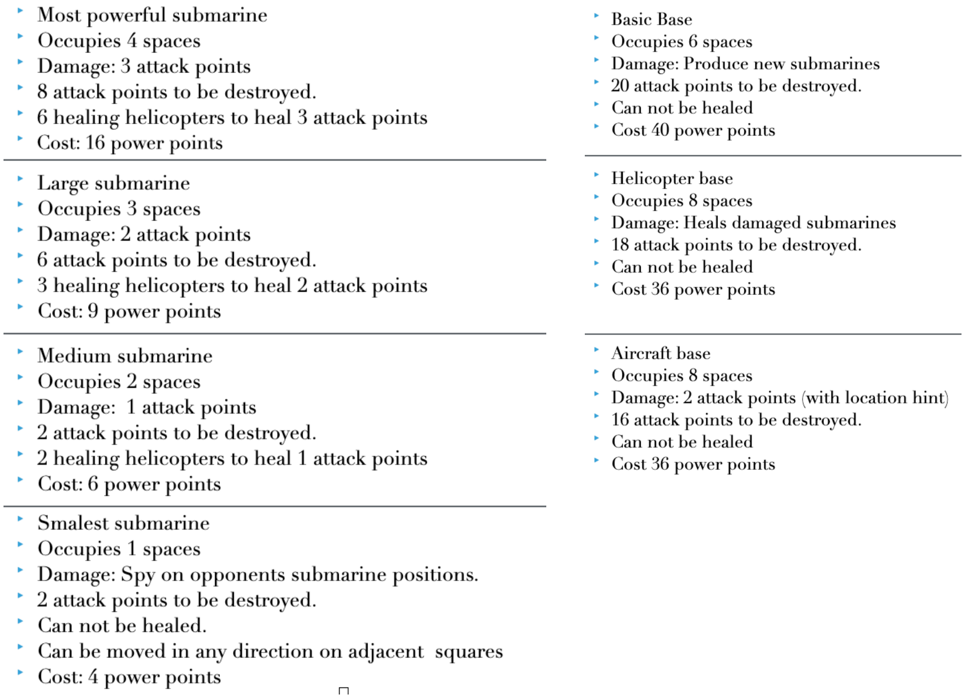
1. Features to be tested / not to be tested

Tests needed to be doe to each class object we have in game. Meaning that object behave the way it designed.

Board tests:

- Can place new objects on the board
- Only 7 objects per player
- No sheep parts outside the board edge
- No diagonal placement
- No overlapping
- No placing objects on top one another.

Below lists represent requirements for each of the fleet objects we have:



- Most powerful submarine
- Occupies 4 spaces
- Damage: 3 attack points
- 8 attack points to be destroyed.
- 6 healing helicopters to heal 3 attack points
- Cost: 16 power points

- Large submarine
- Occupies 3 spaces
- Damage: 2 attack points
- 6 attack points to be destroyed.
- 3 healing helicopters to heal 2 attack points
- Cost: 9 power points

- Medium submarine
- Occupies 2 spaces
- Damage: 1 attack points
- 2 attack points to be destroyed.
- 2 healing helicopters to heal 1 attack points
- Cost: 6 power points

- Smalest submarine
- Occupies 1 spaces
- Damage: Spy on opponents submarine positions.
- 2 attack points to be destroyed.
- Can not be healed.
- Can be moved in any direction on adjacent squares
- Cost: 4 power points

- Basic Base
- Occupies 6 spaces
- Damage: Produce new submarines
- 20 attack points to be destroyed.
- Can not be healed
- Cost 40 power points

- Helicopter base
- Occupies 8 spaces
- Damage: Heals damaged submarines
- 18 attack points to be destroyed.
- Can not be healed
- Cost 36 power points

- Aircraft base
- Occupies 8 spaces
- Damage: 2 attack points (with location hint)
- 16 attack points to be destroyed.
- Can not be healed
- Cost 36 power points

2. Pass/Fail Criteria

Pass criteria: satisfies test statement with no exception

Fail criteria: does not satisfy test cases

3. Approach

Proactive - An approach in which the test design process is initiated as early as possible in order to find and fix the defects before the build is created.

4. Suspension and resumption

National holidays.

5. Testing materials (hardware / software requirements)

Testing software such as Leapwork, T-Plan

6. Test cases

7. Testing schedule

Testa are done synchronously with every milestone in the project.

V. Project Issues

1. Open Issues

After carful consideration out biggest issue is remained cost and budgeting.

2. Off-the-Shelf Solutions

Online money raising services.

2.a.Ready-Made Products

KickStarter can be considered as a source to raise money.

2.b.Reusable Components

4A Engine

Unity

Godot

2.c.Products That Can Be Copied

Battle Lines is an extension of original classic game of Battleship. Our game borrows main rules for the game and implementation idea. Correct copy right must be obtained.

3. New Problems

a. Effects on the Current Environment

Game occupies space on users device.

b. Potential User Problems

Installation and reply should not be a problem along with memory program occupies since the game is all server based.

c. Follow-Up Problems

With the realist of the game we expect to see an average of 1000 users daily visitor our server. With growth of the game and possible expansions migration might be needed. With growth

4. Migration to the New Product

N/A

5. Risks

- Excessive schedule pressure
- Inaccurate cost estimating
- Low quality
- Low productivity
- Cancelled projects

6. Costs

The project require extensive time cost between 3-5 years.

With a developers effort worth of $200 000 per year.

In addition provide all necessary materials and equipment: $10 000

7. Waiting Room

AR version of the game.

8. Ideas for Solutions

There are a lot of good platforms available for game AR game development available on the market.

9. Project Retrospective

The Battle Lines: War of submarines is a complex game that meant to bring together more people. With added rules and board sizes game time is increased and player stakes are higher. Our hope is that simple and intuitive design together with addictive game flow will bring new users daily to play online.

This document summarize all required information on implementation of the project.

The development of the Battle Line: war of submarines is based on strong assumption of the growing strategy games market. The popularity of classic battleship games give us a strong belief that adding new features and expanding into multiplayer concepts will bring attention not only from hands-on users(gamers) but also will attract online game markets to research and invest into the development of the Battle Line: war of submarines.

Educational features of the game can not be underestimated. The game requires problem-solving skills and critical thinking as well as team-oriented skills and communication skills. Taking this fact into account it is assumed that games can be used for educational purposes. The Battle Line: war of submarines game will be created to serve entertaining purpose therefore it is to be associated with leisure and good time. The game design supports users' experience providing exceptional visual and sounds effects. Detailed design gives the user full experience. The Battle Line: war of submarines is a war-themed board game for multiplayer therefore there is a large number of players who are going to play the game at one time. It is assumed that every user will have ethernet access and suitable technology such as smartphones, tablets or laptops.

The game can be a platform for commercial advertising.

VI. Glossary

Attack points: the number that represent the damage made by a fleet object

Power points: the number that represent the cost of a fleet object.

Heal: the terming used to patch the damage made to player's fleet object. This allows user to stay in game longer by fixing their ships.

Hit: when the missile sent by a player gets the coordinate on opponents fleet object correctly without killing it.

Miss: when the missile sent by a player gets the coordinate on opponents fleet object incorrectly.

Sunk: when the missile sent by a player gets the coordinate on opponents fleet object correctly and all squires occupied by the object already been hit.

VII.References / Bibliography

[1]	Robertson and Robertson, Mastering the Requirements Process.
[2]	A. Silberschatz, P. B. Galvin and G. Gagne, Operating System Concepts, Ninth ed., Wiley, 2013.
[3]	J. Bell, "Underwater Archaeological Survey Report Template: A Sample Document for Generating Consistent Professional Reports," Underwater Archaeological Society of Chicago, Chicago, 2012.
[4]	M. Fowler, UML Distilled, Third Edition, Boston: Pearson Education, 2004.
[5]	A website on building software effectively © Martin Fowler
[6]	“Documenting Software Architecture: Documenting Interfaces”, Felix Bachmann, 2002, CMU/SEI-2002-TN-015
[7]	Design Patterns: Elements of Reusable Object-Oriented Software. Erich Gamma, **ISBN-13:** 978-0201633610
[8]	Structure and Interpretation of Computer Programs, 2nd Edition (MIT Electrical Engineering and Computer Science)

VIII.Index