**Software Engineering**

**Group – 5**

Mohammad Asim Iqbal BSCS - 14002

Mahad Saleem BSCS - 14008

Muhammad Murtaza Azam Khan BSCS - 14027

Omer Zaman BSCS - 14046

**Assignment – 3**

**Question – 1: Apply the requirements elicitation techniques you identified in the homework#2 to identify your system requirements**

Stakeholders:

* Gamers

**Scenario – 1**

|  |  |
| --- | --- |
| Start | The user is running an android device with android version 4.2 or above. |
| Normal | User taps the game icon to start the game. Game starts and splash screen is shown. While the splash screen is shown, the game logs in the user to the Google play services account in the background. While the game connects to Google play services and the game loads, a loading bar is also shown which progresses as the game loads. |
| What Can Go Wrong | The android device is not connected to Internet or device’s wifi is off. In this case the splash screen should show an error of connection failure. |
| Ending State | When the loading completes, main menu of the game is shown. |

**Scenario – 2**

|  |  |
| --- | --- |
| Start | The game has finished loading and now the user is in the main menu. |
| Normal | From the main menu, user selects the option to see his/her stats. A new screen opens and user’s stats are displayed. |
| Ending State | The user sees the stats and comes back to the main menu using a button. |

**Scenario – 3**

|  |  |
| --- | --- |
| Start | The game has finished loading and now the user is in the main menu. |
| Normal | From the main menu, user selects the option to see the settings of the game. A new screen opens and user can see options. User can adjust music volume and FX volume. User can also mute music and FX. |
| Ending State | User selects the desired settings and comes back to main menu using a button. |

**Scenario – 4**

|  |  |
| --- | --- |
| Start | The game has finished loading and now the user is in the main menu. |
| Normal | From the main menu, user selects the option to create a deck of cards. A new screen opens. On the top of the screen, six cards are shown which are the battle deck. Below them are the other cards that can be replaced with the cards in the battle deck. In the same screen, user can also tap on the card and upgrade it if the upgrade is available. |
| Ending State | The user creates a battle deck and comes back to the main menu using a button. |

**Scenario – 5**

|  |  |
| --- | --- |
| Start | The game has finished loading and now the user is in the main menu. |
| Normal | From the main menu, user selects the option to start the game. A new screen will appear which will show the progress while match making is done. |
| What Can Go Wrong | The user disconnects from the network or the Internet goes down. In this case, the game should go back to splash screen and load the game again. And the problem of network disconnection remains, an error should be shown to the user about the Internet connectivity. |
| Ending State | A match is successfully made and a new screen opens where the user will play the game. |

**Scenario – 6**

|  |  |
| --- | --- |
| Start | Match making is done and now the user is in the game. |
| Normal | The game starts when both players are connected to the game. Game timer starts and both players start deploying troops. Only those troops can be deployed which consume energy not more than the available energy. After every second the energy bar increments by one. Maximum energy that can be stored is ten units. Both the players deploy troops and the troops fight. |
| What Can Go Wrong | One or both users can lose the connection to the network. In this case, the game won’t stop. When the user connects back to the game, new state of game is loaded on user’s phone. |
| Ending State | The game continues until the time runs out or a user wins the game by destroying opponent’s barracks. |

**Scenario – 7**

|  |  |
| --- | --- |
| Start | Match has been made and the game is started. Now the players are playing the game. |
| Normal | While playing, the player needs to deploy a troop. To deploy a troop, the user will tap on the card he/she wants to deploy and either drag and drop to the certain position or just tap on a certain position of battle field to deploy that troop. Once the player deploys the troop, the troop appears on the screen and does it’s job. |
| What Can Go Wrong | The energy consumption of the troop that has to be deployed is greater than the available energy of the user. In this case the troop won’t be deployed and a toast will be displayed saying that player doesn’t have enough energy. |
| Ending State | Either the troop is deployed and does it’s job, or an error toast shows up to notify user that there is not enough energy. |

**Scenario – 8**

|  |  |
| --- | --- |
| Start | Both players are deploying their troops. |
| Normal | A troop is deployed. If a target is in it’s attacking range, this troop will start the attack on that target. If not, the troop will find the best suitable path and will start traveling on that path. |
| Ending State | The troop keeps walking unit it finds a target. If a target is found, the troop will attack that target. |

**Scenario – 9**

|  |  |
| --- | --- |
| Start | Both players are deploying the troops. |
| Normal | A troop reaches opponent’s barracks. Now the troop will attack the barracks and the defenders in the barracks will also attack the troop. |
| Ending State | Either the troop will die or the barracks will be destroyed. |

**Scenario – 10**

|  |  |
| --- | --- |
| Start | A troop attacked opponent’s barracks and the barracks are completely destroyed. |
| Normal | When the hit points of barracks run out, barracks will be destroyed. This way the player who destroyed the barracks wins the match. |
| Ending State | One of the player wins the match and gets trophies and more cards. |

**Scenario – 11**

|  |  |
| --- | --- |
| Start | Both players are deploying the troops and the game is running. |
| Normal | The game is running. Both players are deploying the troops. Both player’s barracks still have hit points left. Time runs out. When the time runs out and both players are still playing, extra time of three minutes will be given. In this period of time, the energy generation of both players will be doubled. |
| Ending State | The game will continue until the extra time runs out. |

**Scenario – 12**

|  |  |
| --- | --- |
| Start | Extra time is given and both players are still playing. |
| Normal | In this time, if a player destroys opponent’s barracks, the player will win. |
| Ending State | The winner will get trophies and more cards. |

**Scenario – 13**

|  |  |
| --- | --- |
| Start | Extra time is given and both players are still playing. |
| Normal | In the extra time, both players fight and attack, but no one is able to destroy opponent’s barracks. In this case, the game will be draw. |
| Ending State | The game will be draw. No one will get trophies and more cards. |

**Scenario – 14**

|  |  |
| --- | --- |
| Start | The game ends. |
| Normal | The game will give trophies and cards to the winner. And the control will return to the main menu. |
| Ending State | The user comes back in main menu. |

**Question – 2: In addition to the above, answer questions related to all the relevant lenses identified in homework#2**

**Lens of Emotion:**

**What emotions would I like my player to experience? Why?**

I would want the player to enjoy the game and arouse his curiosity along with an element of surprise. That would ensure that the game’s concept is fresh even if the user continually plays it. It would also maintain the level of interest. The game should be able to pacify the user’s enthusiasm and instill a level of satisfaction.

**What emotions are players having when they play now? Why?**

A good experience will leave a lasting impression and hook the user to the game. Putting this in perspective, we will be adding new characters (playable) every three weeks. The user would unlock them without knowing which one, this would bridge the element of surprise and curiosity. A good experience will be had due to the infinite possibilities that would be in play during the battle sequence.

**How can I bridge the gap between the emotions players are having and the emotions I’d like them to have?**

The user would be playing under different circumstances each time due to a great number of character combinations being possible and the nuance in player tactics. This would keep it fresh and maintain the zeal. Also, this would serve as an incentive for the player to return to the game every now and then. The objective is to capture the interest of the player through these emotions. If a user returns and play the game again we’ll be bridging the gap between player and our emotions. New cards, levels, new players, new strategies, etc all these things will capture the attention of the player and will make the player play the game again.

**Lens of Surprise:**

**What will surprise players when they play my game?**

The game will be updated every 3 weeks with new characters. These characters will enable the users to come up with new strategies. The users have no control with respect to which character they unlock. They will be randomly awarded to them after they gain a handsome amount of experience or collectibles in the game. Hence, the game provides various instances of surprise.

**Does the story in my game have surprises? Do the game rules? Does the artwork? The Technology?**

It is a multiplayer game and each time the system selects (applying certain filters) two players to play against each other, the users will be oblivious to the type of characters the opponent is playing with.

**Do your rules give players ways to surprise each other?**

The troops of opponent would unroll only during the battle sequence hence the user would have to adapt to a new situation every time and muster up a new game plan.

**Do your rules give players ways to surprise themselves?**

Every time player wins, new cards will be unlocked which will be a surprise for a user since the user won’t know which cards are going to be unlocked until the player wins a match.

**Lens of Fun:**

**What parts of my game are fun? Why?**

There are many aspects of the game which can be deemed as fun. Playing with your friends online, engaging in a battle for strategic supremacy. Trying different combinations of characters and playing against various other ones is interesting and fun. Allowing players to socialize and send emojis could make the game more entertaining. Adding multiple landscapes (or arenas) would also improve the recreational prospect of the game.

**What parts need to be more fun?**

Playing with friends should be more fun since it can be more competitive and user can discuss strategies with friends and then battle with friends.

**Lens of Curiosity:**

**what questions does my game put into the player’s mind?**

The players will be categorized according to the experience points they have gained and the number of characters (cards) they have unlocked. Players with more experience points will be able to unlock stronger characters or newer ones. This will provide new users with an objective to climb the hierarchy. The user will have to question his strategy or wonder about the new characters that will be introduced. How will the user improvise under strenuous circumstances?

**What am I doing to make them care about these questions?**

The game will constantly provide incentives to the users by offering them new cards in exchange for the game’s currency. This will motivate the new users to continue on their plan.

**What can I do to make them invent even more questions?**

With the introduction of new landscapes which would affect the gameplay, new questions could be invented for the players.

**Lens of Endogenous Value:**

**what is valuable to the players in my game?**

Things valuable to the players are the characters (cards), the experience points, in-game subpar and premium currency.

**How can I make it more valuable to them?**

These can be made more important to the user by offering upgrades for the characters, by rewarding in game currencies on every level up.

**What is the relationship between value in the game and the players’ motivations?**

Allowing players to buy new cards using the in-game currencies. This will motivate the player to level up and accumulate the currencies in order to buy new stuff and for upgrades.

**Lens of Problem Solving:**

**what problems does my game ask the player to solve?**

The game primarily focuses on this aspect. The whole battle sequence is tailored by a set of actions that need to be taken after careful consideration. In order to be tactful, the player must solve the problem of navigating through the opponent’s defense and taking down the base camp. There is no hard and fast rule to cracking this code.

**Are there hidden problems to solve that arise as part of the gameplay?**

Each battle may require a different strategy and skill. There is another problem the user needs to solve, which is coming up with an effective combination of cards. The user needs to develop his deck in accordance to the strategy he intends to follow.

**How can my game generate new problems so that players keep coming back?**

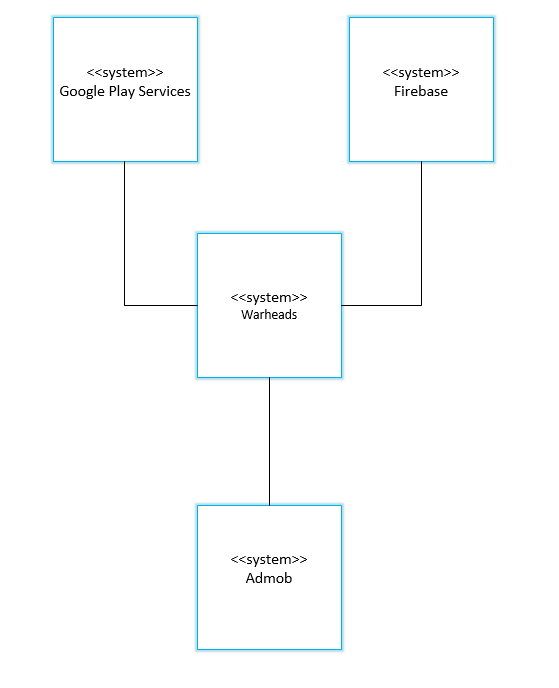
With addition of new cards and unlocking cards, the player must come back frequently and play the game. Addition of new cards means that players have to change their strategy accordingly.

## **Question – 3:** **Identify which of the following modeling techniques are**

## **useful for your project and create those models.**

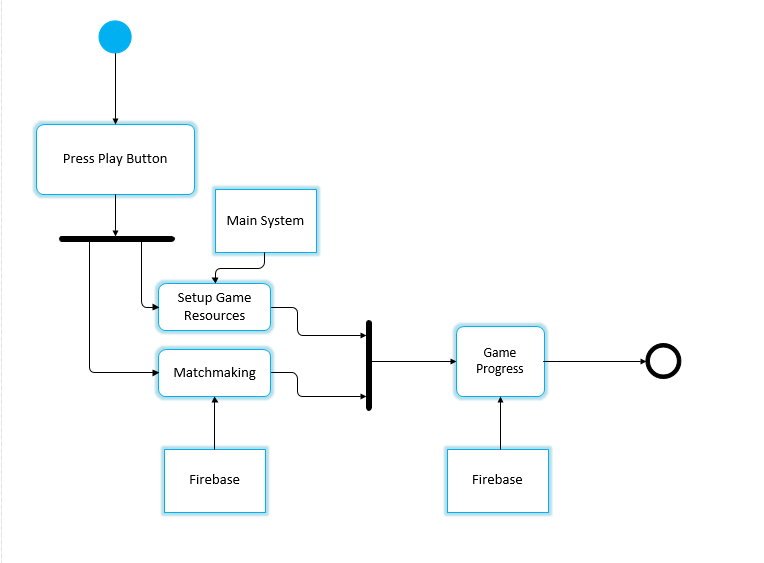
**Context models:**

Context modelling is useful for our software because it will communicate with other systems which lie outside its boundaries. There are three other independent systems that are required for proper functioning of our game. The three systems are shown below in the context model:

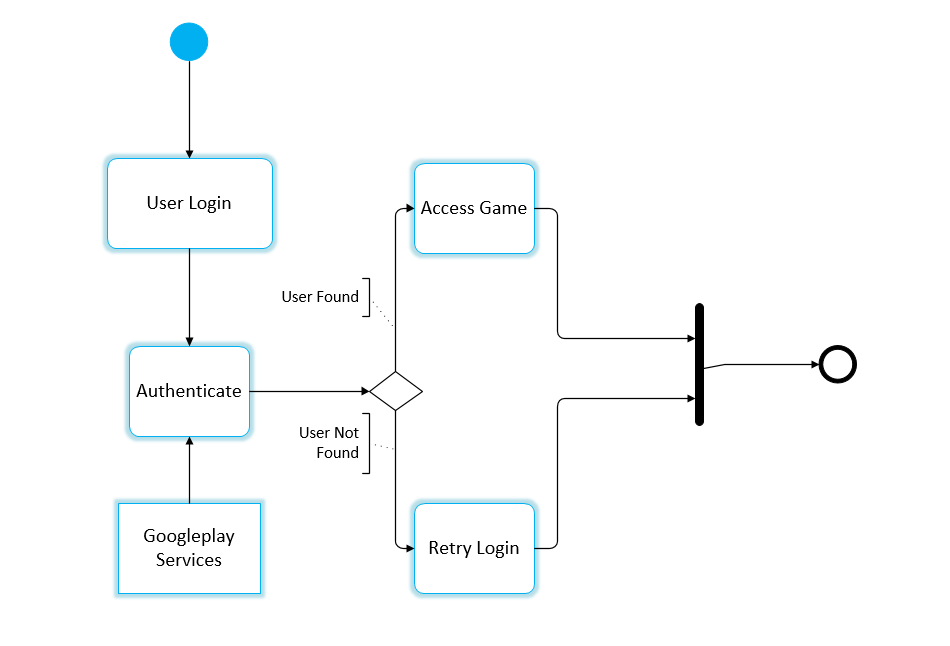


It is important to know how the environment affects our system. To illustrate how our system will function in it, we have developed two process models.

This process model shows the interaction of our system with Firebase:



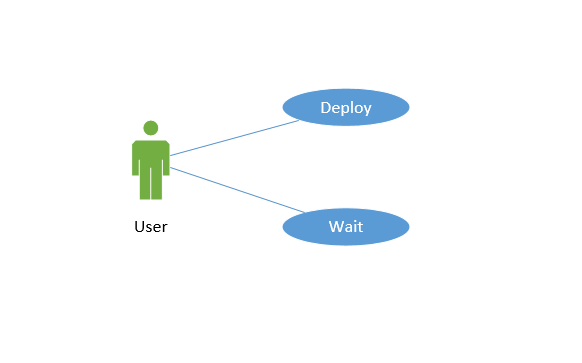
This process model shows our system’s interaction with Googleplay Services:

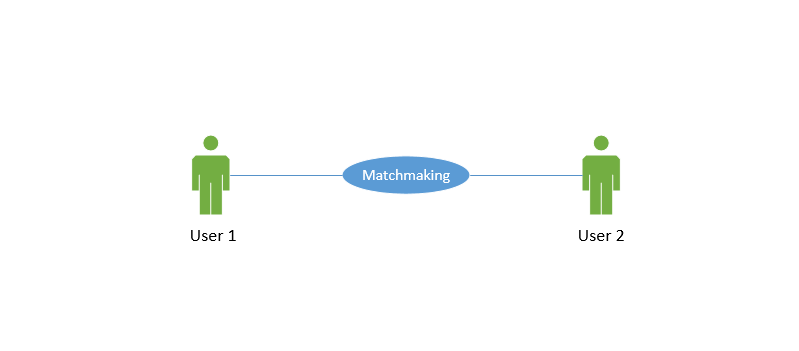


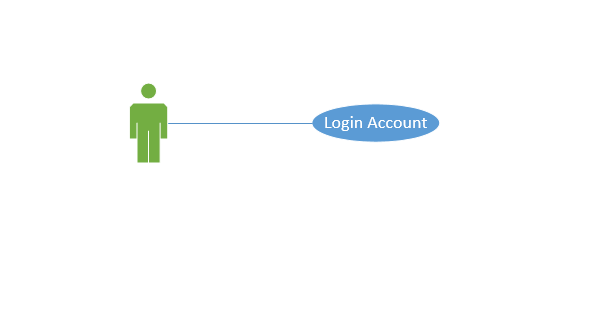
**Interaction Models:**

Our system is an online multiplayer game with various users from different age groups accessing it. Hence, it is important for us to gauge user interaction with our system in order to effectively determine user requirements.

Here are two use cases, that involve user communication with our system:

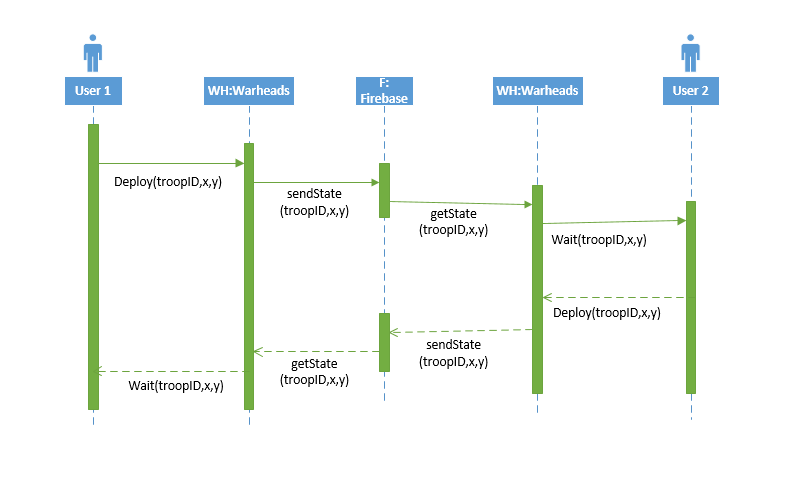




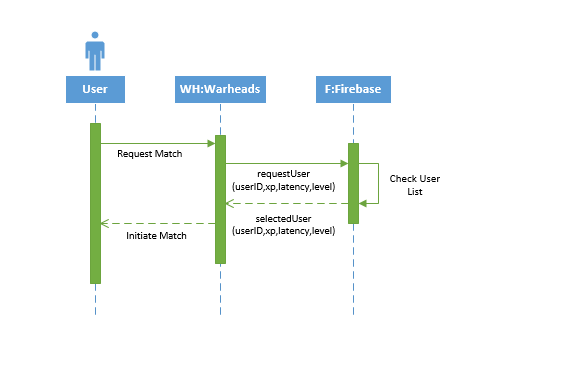


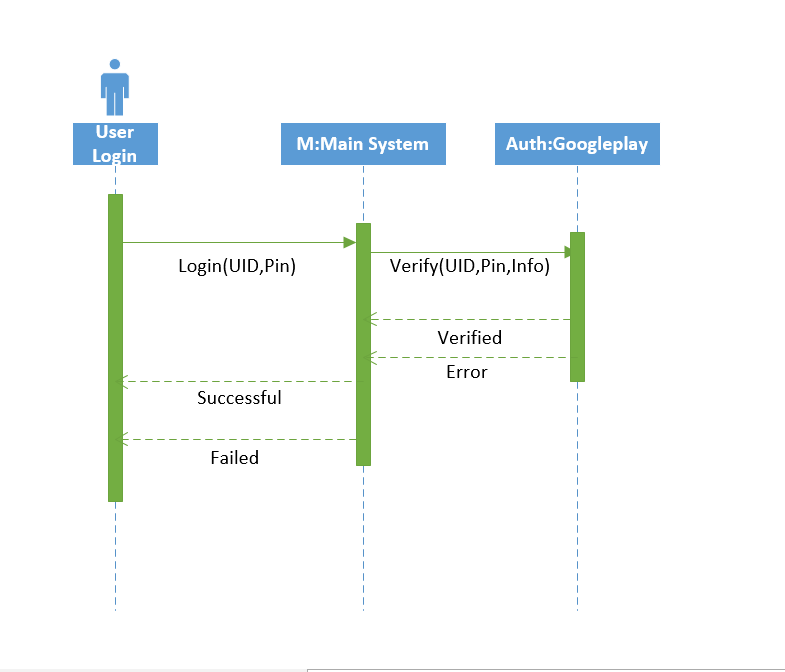
For a more profound elicitation, we have drawn out the actions and the sequences that take place to bring about this interaction.

This is how two users communicate during the game:



This is how two players are paired against each other:

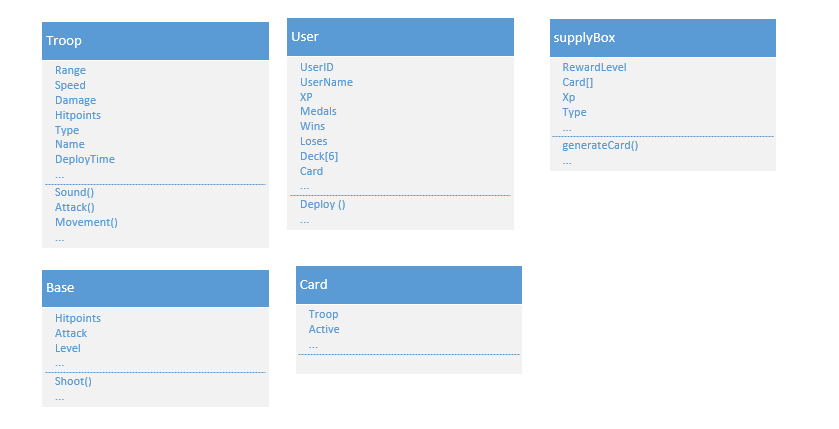
This is how the users log into their accounts:

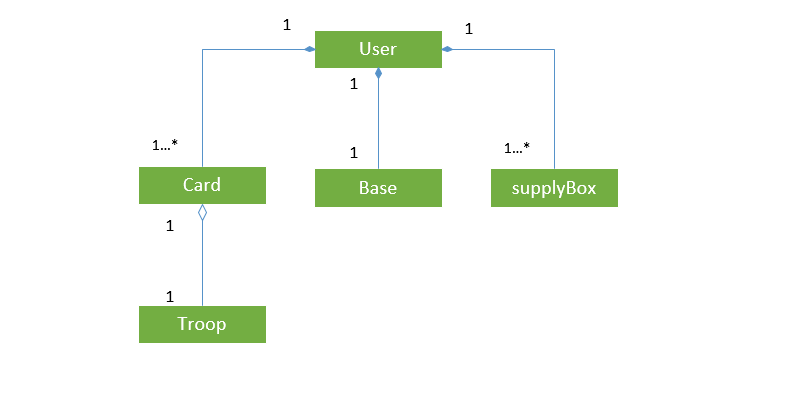


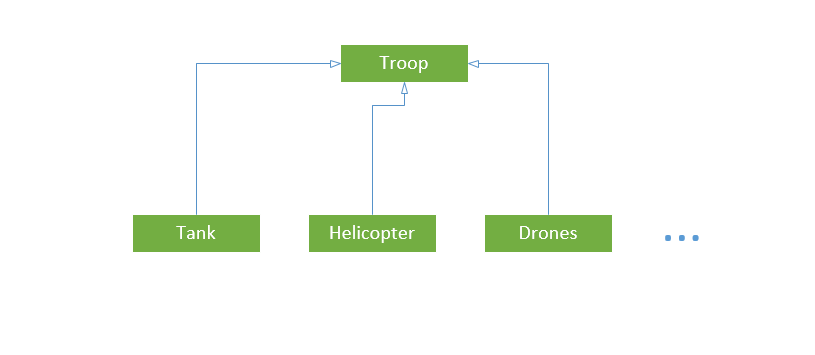
**Structural Models:**

Structured models are important for our system because we will create various objects in our system that will handle different aspects of our game. Hence, they will have a relation with each other.

Below are different objects in our game with their attributes and methods (not all of them are mentioned, only the core components):

Below we have an illustration of their relationships:

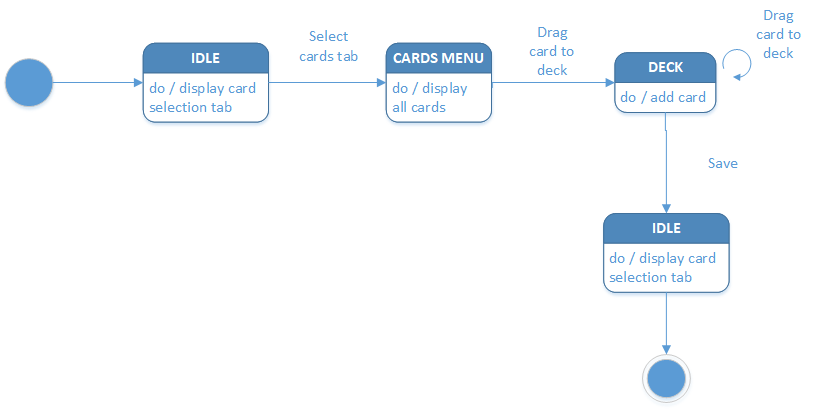




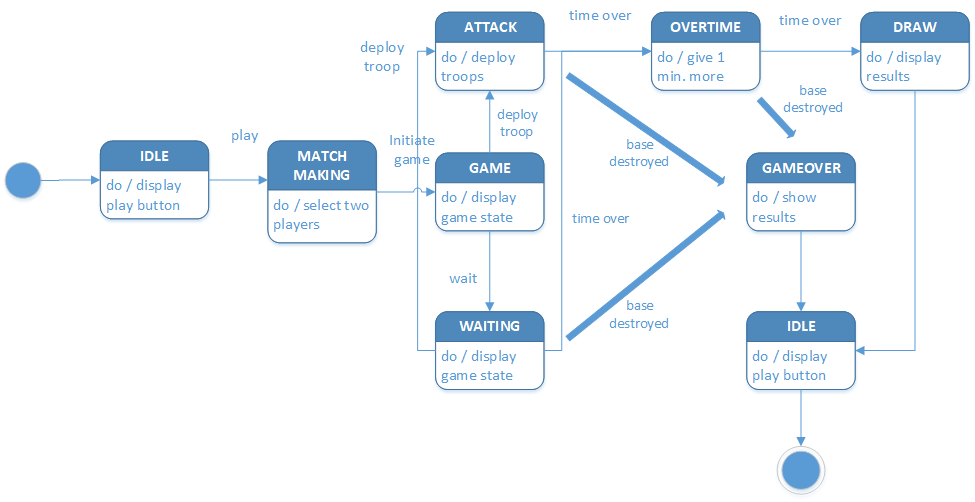
Both aggregation and generalization have been accommodated.

**Behavioral models:**

Our system is mostly event driven. Hence, it is important to look at how the system will respond to different stimuli. Data movement is scarce, so we will ignore the data driven aspect of this modelling technique. We will look at how different events will trigger various actions and events. Which will help us determine how our system should respond to events that occur internally or externally. Below we can see how the deck creation process is driven:



Below we can also see how different events drive our game’s state:



## **Question – 4:** **Identify which of the following lenses are applicable for**

## **your game and address each applicable lens by**

## **answering all the lens related questions:**

**The lens of functional space**

Is the space of this game discrete or continuous?  
The game space is continuous. Every point in the map can open various strategies and vantage points for different troops in the game. Although the map is divided into a grid of 36 rows and 18 columns, this does not mean that the game space is discrete. The grid only defines discrete locations where a player may deploy their troops. Once a troop is deployed, the map is a continuous plane for it and it can go through various states such as run straight to the enemy base, engage with enemies in its range etc.

How many dimensions does it have?The main menu will be 2D whereas the game itself will be 2.5D. This means that the player movement will be in the x-y plane but the characters themselves will be designed such that some depth perception will be apparent to the player. For example, players will be able to tell which troop is taller than the others.

What are the boundaries of the space?  
The game boundary will be the screen space itself. No troops will be able to cross the horizontal edges of the screen (x-axis). Some troops will stay within the vertical boundaries of the screen (y-axis). Some troops, such as the bomber, will fly across from the bottom of the screen to the top of the screen and then out of view.

Are there subspaces? How are they connected?  
Our troops are divided into various categories, such as melee vs. ranged, ground only vs. air + ground, single target vs. area attack. These categories also mark sub spaces for the troops. For example, the presence of an area attack troop who has ranged attack will mark a boundary around that troop. The troop will be able to damage all enemy troops and buildings within this boundary.

**The lens of time**

What determines the length of my gameplay activities?  
The gameplay length is determined by two factors. The first factor is the enemy base tower. When a player destroys the enemy base tower, the game finishes. The second factor is the countdown timer that marks the end of the battle. This countdown timer is further divided into two categories. The first timer is three minutes long. The second timer will be one minute long. If either of the players are unable to destroy the enemy base tower within the first timer, a second timer is initiated. Both players will have to inflict as much damage to the enemy tower within the second timer. The player who either completely destroys the enemy base or does the most damage to the enemy base within the second timer wins the match.

Are my player frustrated because the game ends too early?  
This question is subjective. Gauging from gameplay experience from similar games, the three-minute timer is sufficient for players to execute multiple strategies for success. The player may feel frustrated if they lose too early in the match but this can be remedied by learning different strategies.

Are my players bored because the game goes on too long?  
Our game is a live multiplayer game. This makes the gameplay very dynamic and versatile since there are numerous gameplay styles that players may have. On top of that, the timer implemented in the game play ensures that a high excitement level is maintained throughout the game making the gameplay very enjoyable.

Would clocks or races make my gameplay more exciting?  
We will display the timer clock in our game, as the timer is an integral part of the game not only for creating excitement but also for the rules of the game. It is also possible to view the game as a race since the winner is determined by whoever destroys the enemy base first.

Would a hierarchy of time structures help my games?  
In our game, short rounds are not necessary. The whole objective is to decide a winner within the given time limit. Although it can be argued that adding rounds gives a chance to players who made a single mistake in the game that cost them the match, our approach is that players play their top game in the match and only get one chance to win the match. If they lose, they can always try another match to test their skills.

**The lens of state machine**

What are the objects in my game?  
UI Objects (these objects have generic attributes unless otherwise stated in the next section):  
 Canvas  
 Panels  
 Buttons  
 Toggles  
 Labels  
 Images  
 Vertical Scrollbars   
  
Game Objects:  
 Cards  
 Troops  
 Buildings  
 Map

What are the attributes of the objects?  
UI Objects:  
 Each button and toggle will have its own action listener associated with it that will perform specific tasks.

Game Objects:  
 *Cards* will be visually represented as UI buttons but their role is to allow the user to deploy the associated troop on the map. Each card will have an energy cost that determines how much of the energy resource is used to deploy that troop. Each card will also determine the region of the map where a troop may be deployed, for example, some troops can only be deployed on the player’s side of the map, some troops can be deployed anywhere within the map.

*Troops* are divided into multiple categories:  
 Unit versus building  
 Melee attack versus ranged attack  
 Single target versus area damage  
 Ground only versus air and ground attack  
 Building destroyer versus building and troop destroyer  
 Depending on the categories a troop or building belongs to, it may have a combination of the following attributes:  
 Hit points  
 Damage  
 Damage per second  
 Movement speed  
 Attack speed  
 Target category  
 Range  
 Deploy time (how long it takes for the troop to start taking action after it has been deployed)  
 Engaged (a Boolean value that determines whether a troop is engaged with the enemy or is in a state of moving towards the enemy base)

What are the possible states for each attribute?  
All troop and building attributes except “Engaged” and “Target category” are floats and change when a troop or building is upgraded. Otherwise, they are constant values. “Engaged” is a Boolean that can be true or false. “Target category” is the category that determines what kind of troops this troop engages with.

What triggers the state changes for each attribute?

For troops and buildings, all attributes except “Engaged” and “Target category” are changed when a troop or building is upgraded. “Engaged” depends on “Target category” and changes when any enemy troop from the “Target category” gets within the range of the troop.

**The lens of secrets**

What is known by the game only?  
When a player wins a match, they get a supply drop box as a reward. The supply drop box contains free cards for various troops. A player may collect cards and once they have a predetermined number of cards for a troop, they may use currency to upgrade that troop. The cards that are won from the supply drop box are random and only the game knows what cards will appear in the next drop box.

What is known by all players?  
Players can see the hit points and level of the opponent’s base and troops.  
Players can also see the match timer.

What is known by some or only one player?  
At the beginning of the match, players do not know their opponent’s deck. This is a very crucial aspect of the game as one must be careful in launching the initial attack. For example, if you have only one anti-air troop, it might be a good idea to not use that card in the beginning of the game without seeing the opponent’s strategy.

Would changing who knows what information improve my game in some way?  
We believe that the current game logic maintains a perfect balance of the element of surprise and planned strategy. The current level of knowledge about the game dynamics that the players are equipped with is perfect for this type of game.

**The lens of emergence**

How many verbs do my players have?  
Players can login/log out, view card stats, add card to deck, upgrade a card, view game stats, change various settings, open supply drop box, initiate a battle, select card, wait or deploy cards and quit the game.

How many objects can each verb act on?  
Login/logout does not act on any game object. However, it is triggered through UI buttons.  
View card stats is performed by clicking on cards on the cards screen.  
Add card to deck is performed by clicking the “Use” button on the card stats screen.  
Upgrade a card can be done by clicking on upgrade in the card stats screen.  
View game stats is done by clicking on the player profile button on the home screen.  
Settings for music, fx as well as the help documentation can be accessed through the settings button on the main screen.  
Supply drop boxes can be opened by selecting the available and unlocked supply boxes on the bottom of the main screen.  
Battle can be initiated by clicking on the “Battle” button on the main screen.  
During a match, user can select a card by clicking on one of the three card buttons displayed on a dock at the bottom of the match screen.  
During a match, a user can wait before using one of the three cards displayed in a dock at the bottom of the match screen.  
During a match, a user can deploy troops, at legal positions on the map by using one of the three cards displayed in a dock at the bottom of the match screen.  
A user can quit the game by clicking on the options button on the top right of the match screen followed by the “Quit” button (denoted by a X).

How many ways can players achieve their goals?  
The main objective of the game is to destroy the enemy base with in the first three minutes of the game or by doing maximum damage or completely destroying the base in the extra one minute timer after the first three minutes have expired. To do that, players will need to manage their resource consumption and demonstrate efficient troop deployment.  
In terms of strategies, this game will offer numerous strategies through combinations of units in a deck for a versatile gameplay. Keeping a good combination of troops belonging to different categories is the key to success.

How many subjects do the players control?  
Players will have control over six troop cards out of all the unlocked cards the players have. The match screen will consist of a map, logically divided into a grid of 36 rows and 18 columns. During a match, players decide where to deploy a troop by dragging the associated card on one of the cells on the map.

How do side-effects change constraints?  
This does not apply as there no side-effects.

**The lens of action**

What are the basic actions in my game?  
The actions are the same as described in the verbs.

What are the strategic actions?  
Strategic actions consist of:  
 Choosing when to play the right card.  
 Card placement.  
 The composition of a balanced deck.  
 Efficient management of energy.  
 Predicting the opponents’ moves.  
 Remembering the opponents’ cards.

What strategic actions would I like to see? How can I change my game in order to make those possible?  
Not applicable.

What actions do players wish they could do in my game that they cannot? Can I somehow enable these, either as basic or strategic actions?  
Players would like to be able to control troop movement after deploying them but that is not permitted in the game. Players would also like the ability to see the opponents’ energy but that is also not permitted. Players would also like to chat with their opponents but that functionality is not available.

**The lens of goals**

What is the ultimate goal of my game?  
The objective of this game is to provide a fun filled experience for the players by engaging them in intense strategic battles to be the top rated player. Players gain experience, unlock new cards and upgrade existing ones using the currency gained in the spoils of the battle. Winner takes all.

Is that goal clear to players?  
The goal is made clear to the player through engaging them in tutorial matches as they start the game.

If there is a series of goals, do the players understand that?  
One goal is to win battles. By doing so, players gain experience and earn medals to advance in the battalions, moving them up in the rankings. The tutorials, rewards system and the player stats display makes this intuitive for the player.

Are the different goals related to each other in a meaningful way?  
The goals are related to each other in principle. One goal serves as a step towards reaching the bigger goal.

Are the goals concrete, achievable and rewarding?  
Yes, the goals are concrete and achievable. The matchmaking ensures that both players are evenly matched in experience and medals, making the game fair. As the players reach higher levels, the rewards from the supply drop box also increase.

Does the game have good balance of short and long-term goals?  
Yes. The different levels of goals are well connected and properly balanced.

Do players have a chance to decide on their own goals?  
No.

**The lens of skill**

What skills does my game require from the player?  
Strategic thinking, timing, analytical skills and patience.

Are there categories of skills that this game is missing?  
This game does not require accuracy, or reflexes.

Which skills are dominant?  
Strategic thinking.

Are some players much better at these skills than others? Does this make the game feel unfair?  
Not much better but slightly, yes. It is not something that the player cannot overcome by finding the right deck for them and practising battles.

Can players improve their skills with practice, leading to a feeling of mastery?  
Yes, it is probable as when you play a card, you learn its advantages and weakness and the various ways you can use the card leading to the mastery of that card.

Does this game demand the right level of skills?  
As this game demands the common phenomenon of efficient management of resources, the game will always demand the right level of skills. As the players engage in more and more battles, they improve their skills hence ensuring that they will always get a good challenge.

**The lens of chance**

What in my game is truly random? What parts just feel random?  
The order in which you get 3 of the six cards of your match deck and the opponent’s deck will be truly random that cannot be accurately predicted.  
The rewards from the supply drop boxes may appear random to the player but will be efficiently calculated according to the player’s experience level.

Does the randomness give the players positive feelings of excitement and challenge, or does it give them negative feelings of helplessness and lack of control?  
The randomness of the cards rewarded through the supply drop boxes will raise a balance of positive and negative feelings, with the players always wanting more.

Do players have the opportunity to take interesting risks in the game?  
Yes. Although conservation of resources is key to success in the game, the player may try to go on the offensive quickly in an attempt to finish the game early.

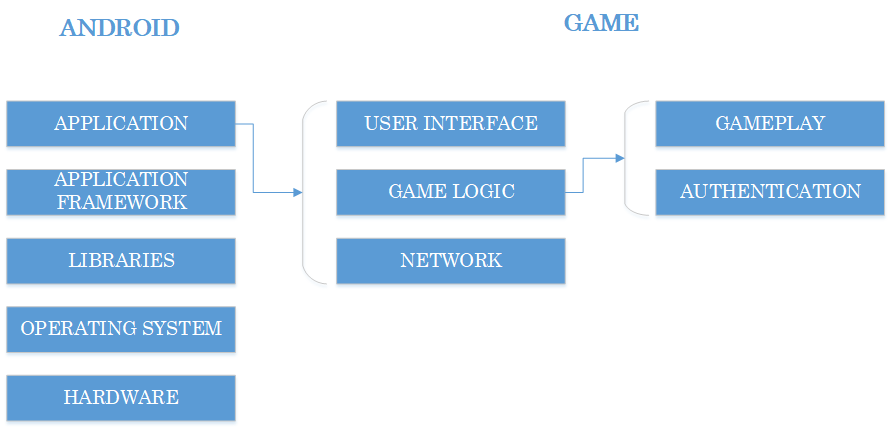
What is the relationship between chance and skill in my game?  
A player might get a bad draw of the first 3 cards but with the right skills, they can still use it to their advantage. It is also possible to get matched up against an opponent who has a solid counter of your deck but it is still possible to turn the game around with the right level of skilled strategic decisions.

## **Question – 5:** **Create architecture diagrams of your system from at**

## **least three of the following perspectives.**

**LAYERED ACHITECTURE**

Following is the layered architectural diagram for our software system.



**Advantages:**

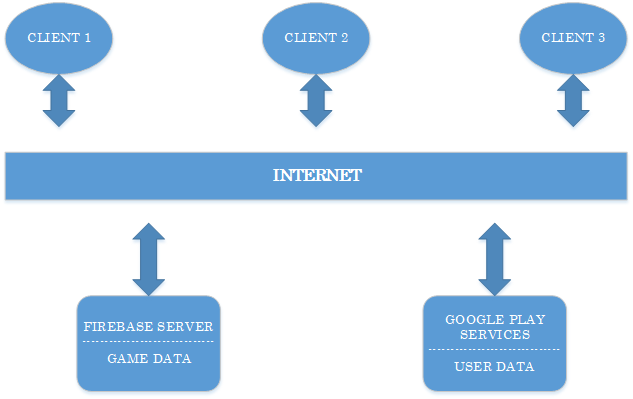
* As we are following modular approach, layered architecture gives a clear understanding of how many major modules our system will have.
* Separation of concerns among layers will help in achieving flexibility and scalability.
* As we are reusing the component of network for authentication and gameplay, layered architecture enable us to do so.
* It will help us to work in parallel for building up the system.
* It will help us to deploy different components independently.
* Testing of each layer becomes easy.

**Disadvantages:**

* There can be an extra overhead for performance if we want a layer to communicate directly. For example, in our system, during gameplay, we want our system to directly communicate to network layer, we might have to bypass authentication layer.
* If we change our network layer, we might need to change our game logic too.

**CLIENT-SERVER ARCHITECTURE**

Following is the client-server architectural diagram for our software system.



**Advantages:**

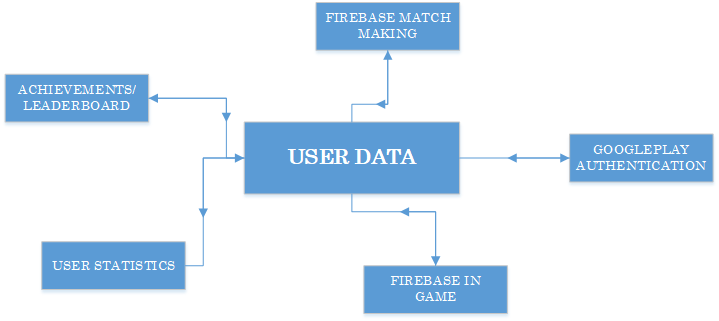
* As we are using multiple services in our system, client-server architecture gives is the best possible approach we would follow. Service of Firebase and Google Play will be available to all the clients.
* If we want to make changes in Firebase protocol or Google Play authentication, we will not have to change our client. The systems are independent.
* Our system will be faster as all the processing is done on client side and servers are used to just retrieve or send data.
* It will help us maintain security as only authorized client will be allowed to access data.

**Disadvantages:**

* If there are frequent simultaneous request from clients, servers might get overloaded, forming traffic congestion.
* Ass the system is centralized, if a server fails, requests from clients will not be entertained.

**REPOSITORY ARCHITECTURE**

Following is the repository architectural diagram for our software system.



**Advantages:**

* Data integrity is maintained and all the components share the same data.
* Different software components are independent from each other and they don’t need to know about each other.

**Disadvantages:**

* As there is not so much data in our system, this architecture might not be the best possible approach to follow.
* All the system components will need to follow same data structure as that of repository. In our system, multiple data structures are implemented.
* If duplication occurs in the repository, all the components get affected.
* If the data gets falsified, all the components will have wrong data.
* Our data is on different distributed systems, handling it in a single repository is difficult.