Bilkent University Department of Computer Engineering

CS 319 Object - Oriented Software Engineering Project

Mafia:TCoS - Mafia:The City of Sin

Analysis Report(Final)

Project Group 2.A:

Başak Melis Öcal, Gökcan Değirmenci, Gökberk Aktulay, Sinan Öndül

Supervisor & Course Instructor: **Prof. Dr. Uğur Doğrusöz**

Submitted at March 11, 2017

Table of Contents

1. Introduction

As a Group 2A, we have decided to build a desktop game, written in Java. Our Game is "Mafia: The City of Sin", with its basic RPG features is a text-based game that is a combined version of well-known browser-based MMORPG "Omerta" and the "Clicking Bad". The game is based on the 1930's Mafia World and the players role-play a gangster within that time. We have been designing our game with multiple features and functionalities which are based on relatively smart calculations(algorithms). Our game differs from Omerta that,in Omerta you are able to kill other players with your power but in Mafia:TCoS that is not possible. Our game has automated drug production rate, money laundering, different imprisonment logic, different end-game logic and much more differences. We are only inspired by Omerta, not copying them.

This first Analysis Report of ours contains a detailed overview of the game which describes the mechanics and the properties of the game. Then it tries to describe the detailed functional requirements, non-functional requirements, use-cases, use-case diagram and several in-game screenshots and more. Hope you like it!

2. Overview

"**Mafia: The City of Sin**", is a single text-based strategy game that players intend to complete the tasks based on committing different sets of crimes aiming a rank increase. A 9x9 map will be used as a kind of main menu that provides submenus for players to choose different set of crimes, trading options, tools to buy etc. Some part of the map will be used as a safe-house for players to observe their own manufacturing/distribution tools, garage, weapons and upgrades. All of the crime options will include different chance and health risk percentages based on the rank of the player. Players will be able to buy upgrades depending on their money gained from crimes.

"**Mafia: TCoS**" is a desktop game which will be controlled by mouse. Additional keyboard controls can be added throughout the development process.

2.1.   Gameplay and Control

Only the mouse is needed for playing the game. As mentioned earlier, players will be able to choose submenus from the map by a single click. All of the radio button options in the submenus and other buttons will be controlled by the mouse. The ESC key causes the game to close.

2.2.   Map

A 9x9 grid will be used as a map which provides different labels for crimes, car theft, narcotics etc. in each grid.
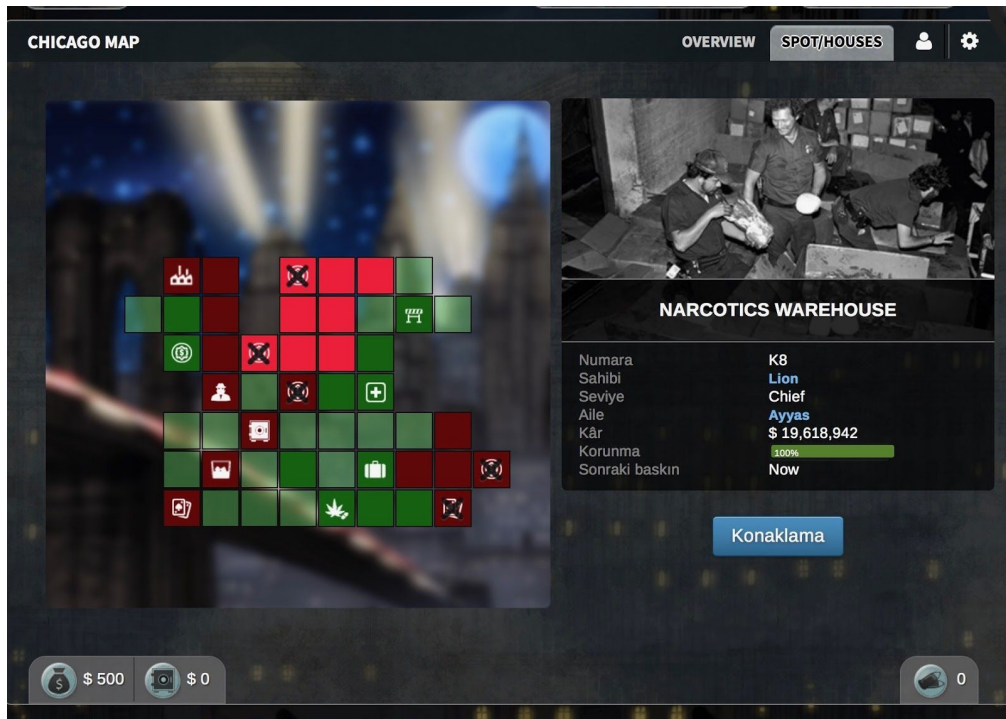

Figure-1: 9x9 Grid[1]

2.2.1. Crime Spots

Labels with specific symbols for crime, car theft, smuggling, car and weapon selling/ buying will be distributed to the map. Players will be able to choose submenus with a single click.

2.2.2. Safe House

A 4x4 grid will be reserved for each player's safehouse. Goods owned such as cars, weapons, bullets, narcotics manufacturing/distribution tools and laundering mechanisms will be observed from the specific grids in the safe house. Number of each property will be written and upgrades will be shown with number of stars in each grid.

2.2.3. Submenus

Whenever a label is clicked, a submenu appears on the right half of the screen with a set of options. For "Crime" and "Car Theft" submenus, chance of success and health risk percentages will be provided in addition to the task name. Submenus for selling/buying involve the amount of the product to sell or buy that user can manipulate to change the amount. "Smuggling" submenu involves another submenu for manufacturing, distribution and laundering.

2.3. Crime Committing

2.3.1. Crimes

When crimes label is clicked, a submenu appears on the right half of the screen with a set of crime committing options. Each option involves task name, chance of success and health risk percentages. Percentages are calculated with an algorithm based on rank, weapon and bullet amount of the player. By clicking "Go for it!" button, players will be able to commit crimes. Each attempt results in either a success, jail time, or fleeing without a success depending on the success percentage of the crime. If the player commits the crime successfully, his/her FBI raid chance will increase by 3% with also an increase in rank and money.

2.3.2. Car Theft

Car theft submenu appears on the right of the map, when related label is clicked from the map. Submenu involves a variety of options, in addition to chance of success and health risk percentages written next to each option. Again percentages are calculated considering the rank, weapon and bullet amount of the player. By clicking "Go for it!" button, players will be able to make an attempt. Attempt may result in a success, jail time or fleeing away without success. If a success occurs, FBI raid chance will increase by a percentage of 3 with an increase in rank. The stolen car will be added to player's inventory by also updating the garage in the safe house.

2.3.3. Smuggling

When "Smuggling" label is clicked, a submenu appears on the right half of the screen, as any other crime option. Smuggling has 3 major submenus; manufacturing, distribution and laundering. A variety of tools with different prices and FBI raid chances will be provided to the player for manufacturing and distribution. Inventory will be updated for tools bought or sold. Drug manufacturing and distribution also have options for updating tools for greater manufacturing and distribution. 3rd submenu, "Money Laundering" involves safe business options. User will be able to adjust the FBI raid chance by mainly using this option. Laundered money will be a counter based on the player's safe businesses. Drugs produced will be shown on the inventory screen as a counter.

2.4. Tiredness

After completing each crime or car theft attempt, players have to wait for 90-120 seconds for a new task from the same crime category. While waiting players are able to complete another task from a different category.

2.5. Jail Time

 Players may end up in jail due to crime committing or car theft attempts depending on the chance of success. Also, possibility of FBI raid over %50 may result in jail time. Time in jail, disables the player actions for precalculated time. Game ends when player ends up in jail for 20 times.

2.6. FBI Raid Chance

FBI raid chance increases depending on crime committing, car stealing, buying drug manufacturing/distribution tools. It can be decreased by laundering money or buying upgrades. Increased FBI raid chance results in either jail time or money loss. Player will be able to continue the game with the laundered money if the money is seized by FBI.

2.6. Upgrades

When "Upgrades" menu is clicked, a submenu appears on the right half of the screen. In this submenu, there will be a variety of options including decreasing DEA raid chance, increasing the manufacturing and distribution rates, upgrades for laundered money, health bonuses and weapons. Upgrades can be bought depending on the money and the player's current rank.

2.7. Ranking System

A rank increase is based on gained experience by completing as many tasks as possible. Whenever player levels up, the title of the new rank appears in his/her account.

2.8 End of The Game

Game ends when all of the health of the player is used up or when the player ends up in jail for 20 times. When game is over, player has to create a new account to start from scratch.

3. Functional Requirements

- **Users should be able to create an account by choosing a nickname and a password.**
  The system shall have proper and clear Signup/Login process. Users should be able to signup and login effortlessly.

- **Users should be able to Play Game.**

  1. Users should be able to make digital money and gain experience with various in-game activities.
  The system designed on player's building their Illegal Mafia Empire by committing crimes, steal a car, gain money and experience.

  2. Users should be able to commit crimes, do drug manufacturing and rob cars by clicking the relevant slots of the 9x9 game map.

  The main game screen contains one big map which shows possible crime options , drug manufacturing and distribution slots. Users be able to click the specific slot inside the map, after that subscreen appears about their selection. (Crime subscreen, Theft-a-car subscreen, etc.)

  3. Users should not be able to commit crimes in succession.

  After every attempt to commit crime or rob a car, users have to wait for some pre-calculated time to attempt again. Otherwise, it would be meaningless to add "commit crime" mechanism.

  4. Users should be able to select between five pre-defined crime options and be able to see his chance to do them successfully.

  The systems' "commit crime" and " rob a car " mechanism and their possibilities are solely based on mathematical calculations on the back-end side. The chances(risks) are calculated by some predefined algorithms that rely on user's current stats such as "rank", "health status", "upgrades they have" and +- %10 randomness.

Figure-2: Committing crimes[1]

5. Users should be able to gain digital money, experience(xp) after every successful crime attempt: unsuccessful attempts could cause the user end up in Prison.

After clicking the "Go for it!" button on the Crime sub-screen, user has got five possible scenarios:

       1- End up in jail (can not do anything, have to wait) + gain xp

       2- Gain money + xp & Lose Health

       3- Gain money + xp

       4- Lose Health & Get away from the cops (not end up in jail)

       5- Get away from the cops (not end up in jail)

6. Users should be able to see multiple parameters/stats about their account.

Users should be able to easily see their current rank status, health, upgrades they have and the possibility of the FBI raid at the top banner of the main screen in realtime.

- **Users should be able to adjust sound effects and mute the background music as if they want**

  Users should be able to mute the background music by simply clicking the Settings tab and clicking the Mute icon.

- **Users should have a profile page and be able change their profile picture by image uploading or may choose from the predefined images.**

Profile pages of the every user designed to be a place where they can add "wise-sayings" , "mottos" and profile pictures. Users should not be able to add contents which are abusive and/or racist.

- **Users should be able to see informations about other players and view their profiles.**

  Users will able to check other players' profile pages and have more

  information about them easily. Also they will be able to see the stats

  accomplished by the owner of that profile.

- **Users should be able to aware of the end of the game and see the total account stats.**

  Users should be able to know the fact that game has been finished. The

  End Screen will be appeared and Credits flows.

- **Users should be able to get help and additional info about the game.**

  Every screen has got a "i" or "?" icon on the top-right corner. This will

  allow users to get info/help every time they stuck.

- **Users should be able to restart the game and delete all of its account's data.**

  If user wants to delete his account or restart the game they should be

  able to do those action from the Settings menu.

- **Users should be able to continue where they left the game earlier.**

  After selecting Load Game option and entering their user credentials,

  users should be able to keep playing the game where they left earlier.

4. Non-Functional Requirements

- Game graphics and the GUI should be designed as clear as possible in

  order to reduce the learning-curve and be more user-friendly.

- Controls of the game should have short response time and will update

  the data fast. Thus enabling user to play with minimal delay.

- Plot of the game should be easily understandable.

- The database connection between the client and the server must be

  reliable and also shall be encrypted.

- Some critical client-server connection mistakes such as "affect the other players by sending data to server from another account" should be avoided.

Pseudo-Functional Requirements

- Only Java and its popular frameworks such as JavaFX, Spring or Play! for the Main Classes (main operations) for Helper Classes (helper operations) shall be used.
- The Database-Client connection may use Server-less approach and therefore use some Cloud APIs.
- Many-to-Many Database Design shall be implemented.
- For mockups, graphical design objects and several illustrations we shall use Adobe Photoshop, Sketch, Adobe Illustrator.

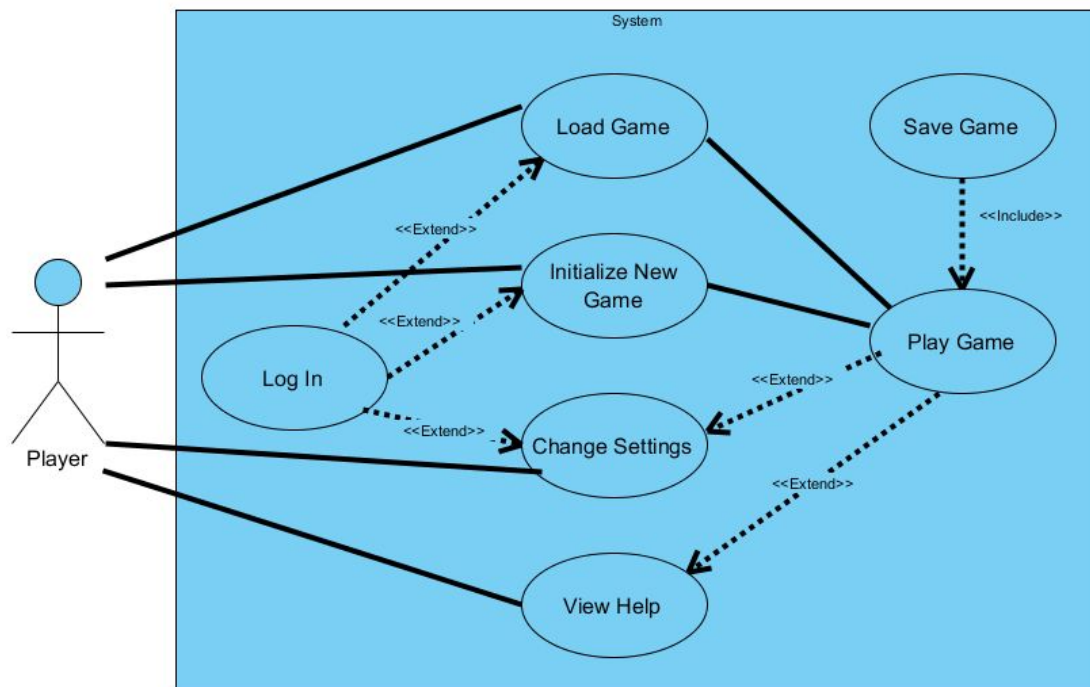# 5. System Models

5.1 Use Case Models

Figure-3: Use Case Diagram

5.1.1 Log In
Use Case Name: Log In
Actor: Player

Entry Condition: Player is not logged in.
Exit Condition: Player is logged in.
Flow of Events:
1. Player enters a name and a password.
2. Player presses log in.
3. System verifies name and password.

Alternative Flow of Events:
4. At any time player can hit cancel to leave login screen.
5. If the password and the name do not match, return to step 1.
6. If player wants to create an account they can press "Sign up".
7. Player enters a name and a password.
8. If the name exists return to step 7.
9. Else system saves the name and the password to a file.

5.1.2 Change Settings

Use Case Name: Change Settings
Actor: Player
Entry Condition: Player has logged in with an account.
Exit Condition: Settings are changed.
Flow of Events:
1. Player makes changes on the available option.
2. Player confirms the changes.
3. System enacts the changes.
4. Player is returned to the previous screen.

Alternative Flow of Events:
5. If player decides not to make any changes press cancel to exit Settings.
6. Player can select "Restore to Default" option to restore the default settings.

5.1.3 View Help

Use Case Name: View Help
Actor: Player
Entry Condition: The program is running.
Exit Condition: Player is satisfied with the explanation.
Flow of Events:
1. System prints a premade guide.
2. Player reads relevant parts.
3. Player presses the close button.
4. Player returned to the previous screen.

5.1.4 Initialise New Game

Use Case Name: Start New Game

Actor: Player

Entry Condition: Player has logged in with an account.

Exit Condition: System successfully initialised the game.

Flow of Events:

1. System creates an instance of the game.
2. System sets the game values to default or random values.
3. Player proceeds to play the game.

5.1.5 Load Game

Use Case Name: Load Game

Actor: Player

Entry Condition: A save file for the Player's account is available.

Exit Condition: Save file is read. System successfully initialised the game.

Flow of Events:

1. System reads the save file.
2. System creates an instance of the game.
3. System sets all the values to the values recorded in the save file.
4. Player proceeds to play the game.

5.1.6 Play Game

Use Case: Play Game

Actor: Player

Entry Condition: An instance of a game is initialised.

Exit Condition: The player runs out of health and dies or goes to prison too many times.

Player selects "Back to the Main Menu" from.

Main Flow of Events:

1. Player starts playing the game.
2. Player click the slots inside the map.
3. Subscreen appears for each clicked slot on right-side of the main screen.
4. Player attempt to commit crimes or theft a car or do drug business.
5. Player gains money and experience .
6. Player tries to get the uppermost rank available + build a Mafia Empire
7. Player loops through the steps 2-3-4-5-6.

Alternative Flow of Events:

8. At any time the player can select "Save Game". Resume the ordinary flow of events.
9. If player goes to prison too many times, system ends the game.
10. If player runs out of health, system ends the game.
11. Player selects "Return to Main Menu", system ends the game.

5.1.7 Save Game

Use Case: Save Game

Actor: Player

Entry Condition: A game is in session.

Exit Condition: A save file is successfully created.

Flow of Events:

1. System creates a file.
2. System writes values to the file.

Alternative Flow of Events:

3. System failed to create a file cannot be created due to external errors.
4. Print an error message.
5. Resume game

5.2 Object and Class Model

The object diagram of "Mafia: TCoS" is illustrated below. Important features of the classes are explained briefly as follows:

AccessManager: Controls access to the game by providing signup and login options. Checks whether the entered username and password match with an

existing account. Also, it creates a new account with the preferred account information.

Account: Stores all the information about the player such as rank, heath, money, inprisonment count etc. which are necessary in game logic.

Profile: Provides customization to the user by profile picture and profile descriptions.

Inventory: Stores what the user has.

GameEngine: It's the main class of of our game which controls most of the game features and does the necessary calculations.

StaticCrime: It's named as static as the objects of this class don't require continuous updates. **CarTheft** and **Crime** classes override some methods from this class.

DynamicCrime: Gets the name from the dynamic counters in its child classes and itself.

ViewManager: Class provides the full screen window and all the graphics. It's the main UI class.

MapManager: Draws the map by filling the slots with labels. Updates the 'safehouse' of the player whenever an update occurs and loads the map when the game is initialized.

SubScreenManager: All of the submenus for different crime types are stored in a multidimensional array based on the rank of the player.

AccessManager
-username : String
-password : String(bcrypt)
-sessionID : long
-session : boolean
+IsValid() : boolean
+login()
+signup()
+record()
+authenticate()
+isSessionExpired()

<<use>>

Account
-rank : String
-totalExperience : double
-health : double
-money : double
-inprisonment : int
-accountID : int
-unlockedAchievements : String[]
+deleteAccount()

Profile
-profilePicture : Image
-profileDescription : String
+uploadProfilePicture() : void
+changeDescription() : void
+deleteProfilePicture() : void
+isUserCreatedProfile() : boolean

1

Inventory
-drug : long
-bullet : long

ViewManager
-view : View
+showProfile()
+showLogin()
+showHomePage()
+showHelp()
+showCredits()
+showError()
+showSignUp()

InputManager
+mouseClicked()

reports user inputs

handles user inputs

provides user information

1

GameEngine
-raidChance : double
-successChance : double
-isOver : boolean
-launderedMoney : double
+calculateRaidChance() : double
+waitForNewCrime() : double
+updateInprisonment()
+updateMoney()
+waitForJailTime()
+calculateSuccessChance()
+updateRank()
+updateExperience()
+updateHealth()
+giveRandomness()
+gameLoop()
+initGame()

SoundManager
-isMusicEnabled : boolean
-music : Clip
-isSoundEnabled : boolean
-sound : Clip
+playMusic()
+adjustVolume()
+playSound()

recieves info from

<<Interface>>
EditableView
-view : View
+draw()
+disableView()
+isViewEnabled()
+showView()

Settings
+update()
+onChangeApplied()

StaticCrime
-isSuccessful : boolean
+tryCommitCrime()
+isCrimeCommitted()

DynamicCrime
-raidChance : double
+purchaseAsset()

MapManager
+loadMap()
+fillSlots()
+updateSafehouse()
+selectLabel()
+showView()

regulates

SubScreenManager
-view : View
-submenu: int[][]
+showView()
+showProfileInfo()

updates

Map
-mapGrids : int[][]
-locX : int
-locY : int
+changeLabelCoordinates()

purchases

CarTheft
+addCarToInventory()
+tryCommitCrime()

Crime
+tryCommitCrime()

<<Interface>>
Asset
-raidChance : double
-price : double
-rankThreshold : int
+checkRequirements()

Manufacturing
-drugRate : long

Laundering
-moneyRate : double
-launderedMoneyRate : double

Distribution
-drugRate : long
-moneyRate : double

*Figure-4: Mafia: Object and Class Diagram

5.3 Dynamic Models

5.3.1 Activity Diagram

The following activity diagram demonstrates the general flow of the game. After the user enters his username and password, saved game is loaded if the authentication is successful. Alternatively, user may sign up, if he/she doesn not have an account. User's account created and game initializes immediately once the sign up occurs. User can select four main options from the game map which are commit crime, car theft, smuggling and upgrades. The first three options may be successful or unsuccessful. Successful case involves a series of upgrades on health, money, rank etc. If the attempt is unsuccessful user is disabled from the game for a predefined period.. When the crime attempt is resulted, "Game Over" situation is checked which requires all the health to be used up or imprisonment for 20 times. If game isn't over, FBI raid is checked and fine is applied. Before getting the input from the user and starting to loop again, system controls that if any achievement unlocked or not. Mafia:TCoS is not a fast-paced, leveled game so its play game mechanics have almost infinite combinations of choices which solely depends on the user's strategy.
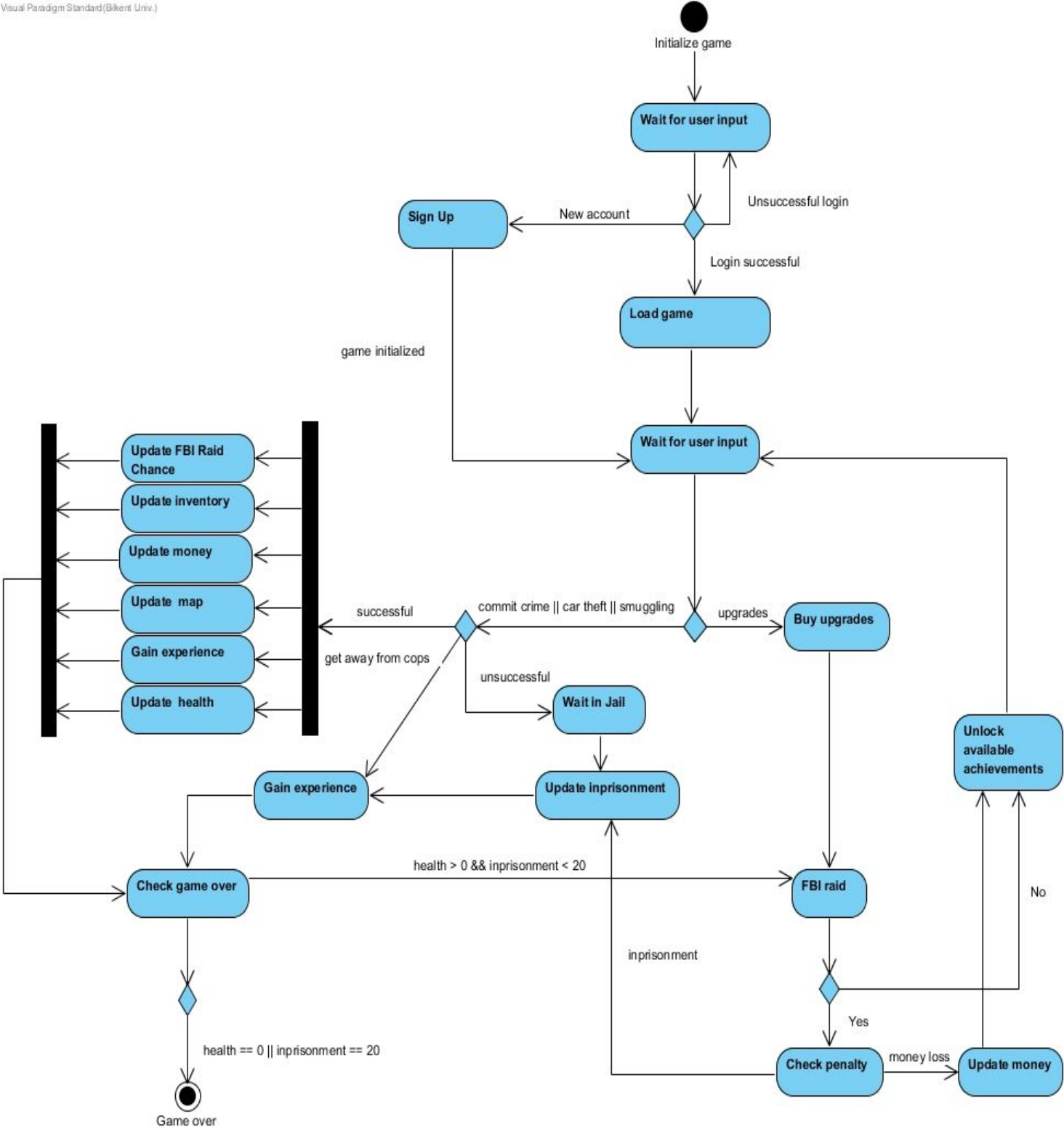
Initialize game

Wait for user input

Unsuccessful login

New account

Sign Up

Login successful

Load game

game initialized

Wait for user input

Update FBI Raid Chance

Update inventory

Update money

Update map

Gain experience

Update health

successful

commit crime || car theft || smuggling

upgrades

Buy upgrades

get away from cops

unsuccessful

Wait in Jail

Gain experience

Update inprisonment

Unlock available achievements

No

Check game over

health > 0 && inprisonment < 20

FBI raid

inprisonment

Yes

health == 0 || inprisonment == 20

Check penalty

money loss

Update money

Game over

Figure-5: Mafia: TCoS Activity Diagram

5.3.2 Sequence Diagrams

Scenario: John is not a registered user of the game. However, he does not know that yet. He remembers that he has been signed up already, tries to enter his mostly used username "*JohnTheSupertramp*" in a username field and his regular password "*123johntst*" in a password field. He tries to login with those credentials. But as we already predict, system shows a error message similar to this: "*You are not a registered user. Please check your username and password. If you have not signed up yet, Signup!*". John presses the **Signup screen** button**.** View Manager of the game displays the Signup screen and waits for John to enter a valid username-password combination. John enters his mostly used user credentials. Access Manager checks the entered credentials with a REGEX rules and also checks availability of the entered username within a single method. This time his password is a valid password according the game's predefined password regex rules and also his username have not been already taken. After that, View Manager makes the actual Signup button visible and clickable. John clicks the signup button. System tries to record his credentials on the database(within an Access Manager). Luckily, the whole process went errorless and his credentials recorded. Lastly, his Account instance has been created with the predefined values(rank=Empty-Suit, experience=0, money = 100.0$, health = 100)
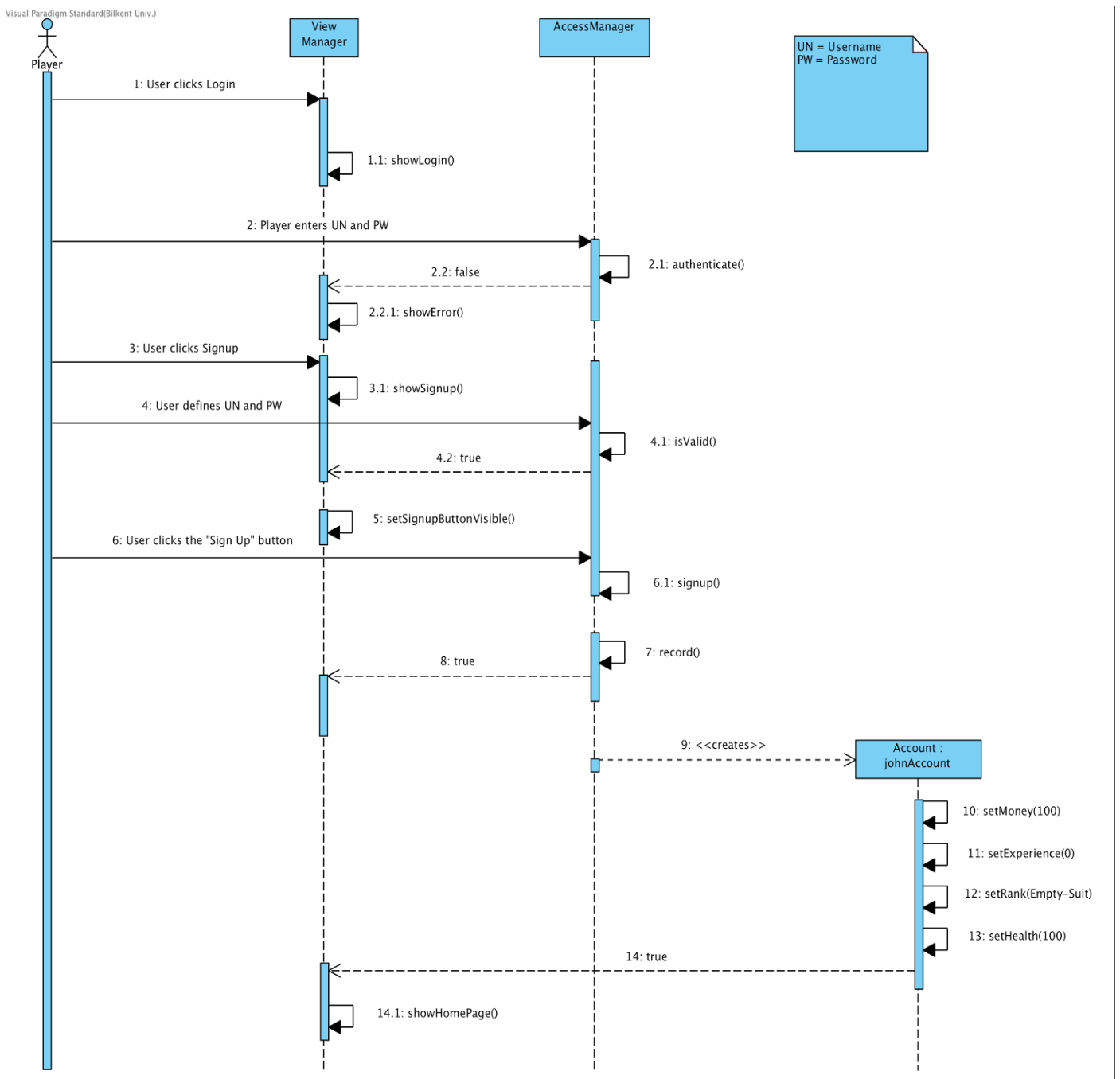
Player

View Manager

AccessManager

UN = Username
PW = Password

1: User clicks Login

1.1: showLogin()

2: Player enters UN and PW

2.1: authenticate()

2.2: false

2.2.1: showError()

3: User clicks Signup

3.1: showSignup()

4: User defines UN and PW

4.1: isValid()

4.2: true

5: setSignupButtonVisible()

6: User clicks the "Sign Up" button

6.1: signup()

7: record()

8: true

9: <<creates>>

Account : johnAccount

10: setMoney(100)

11: setExperience(0)

12: setRank(Empty–Suit)

13: setHealth(100)

14: true

14.1: showHomePage()

Figure-6: Sequence diagram for the Signup/Login & Account creation process

Scenario: Schneizel-sama has been playing the game for a while and has been imprisoned 18 times so far. He proceeds to commit a crime and fails. He goes to prison and that game screen freezes until time is served. He tries to commit a crime and fails once again. Since this is his 20th imprisonment, the game ends for Schneizel-sama and the instance of game is engine destroyed.
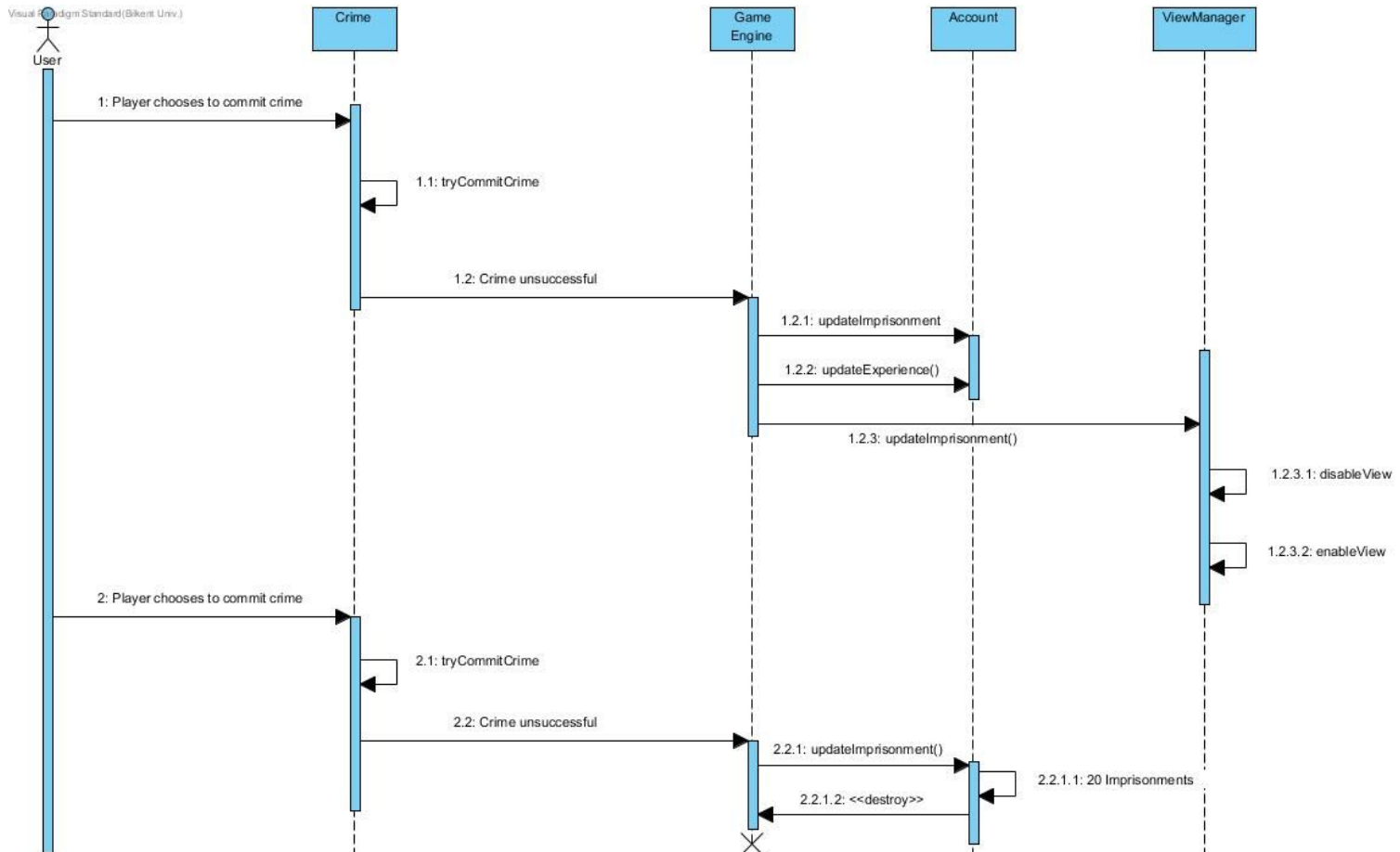
Figure-7: Commit crime example resulting in game over

5.4 User Interface

5.4.1. Navigational Paths
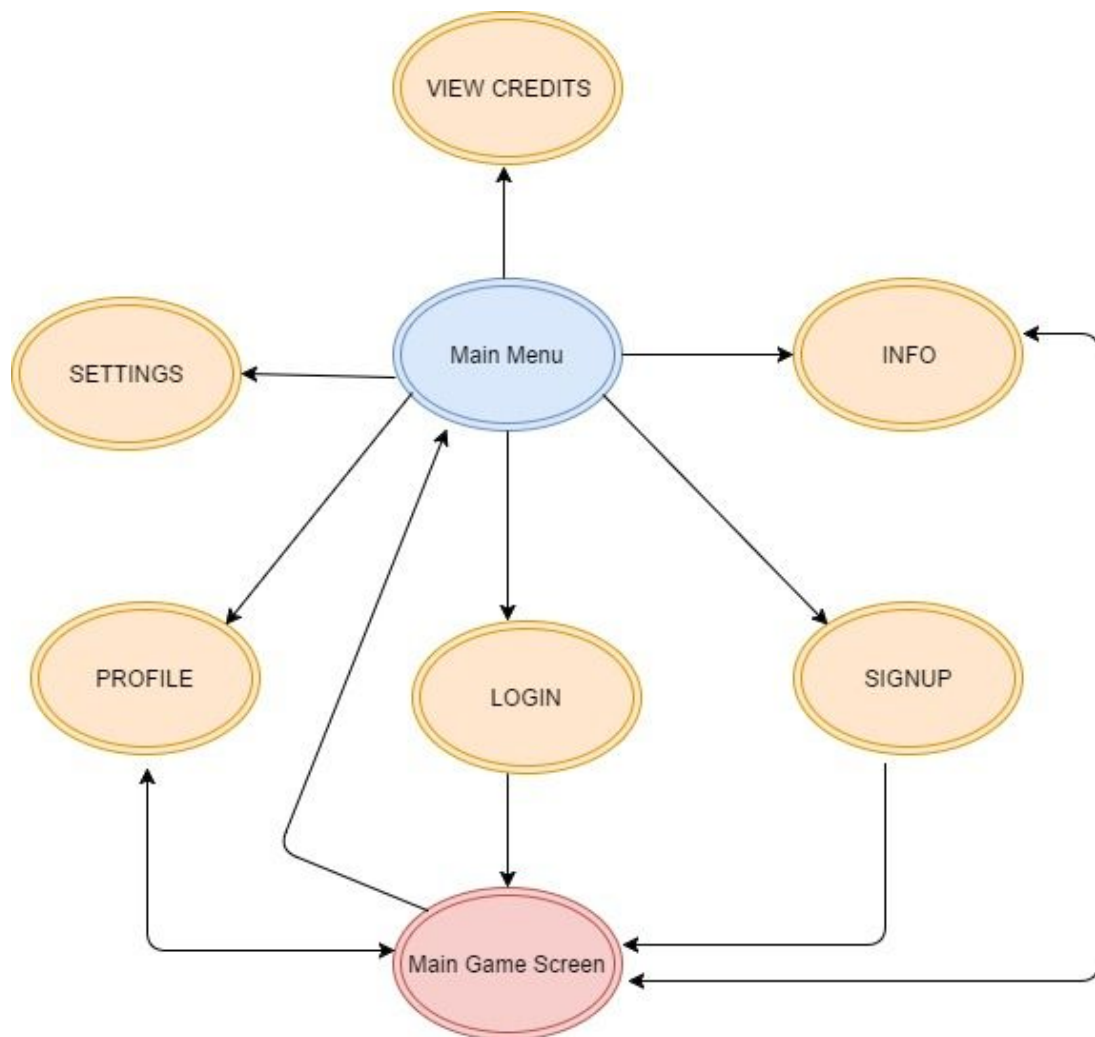
Figure-8: Navigational path of Mafia: TCoS

5.4.2. Screen Mock-ups
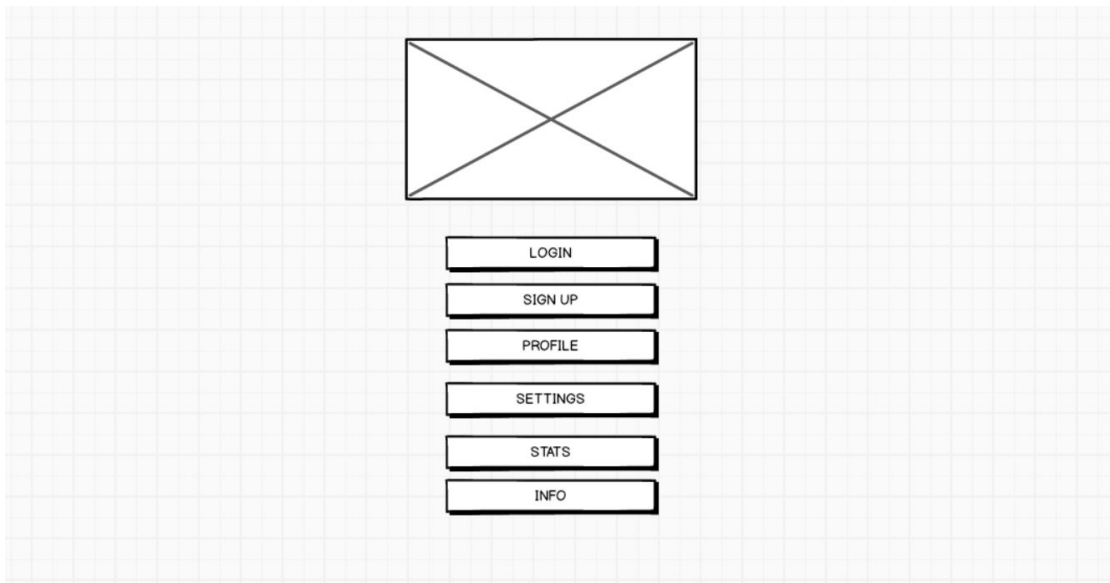
Figure-9: Main menu screen

Main Screen of the game will be constructed as shown in Figure-9. There will be an image of the game at the top, login and signup buttons for users to access their accounts or create a new one. Profile button will open user information. Settings will open settings screen, Stats button will open stats screen and Info will open a screen that shows the game information.
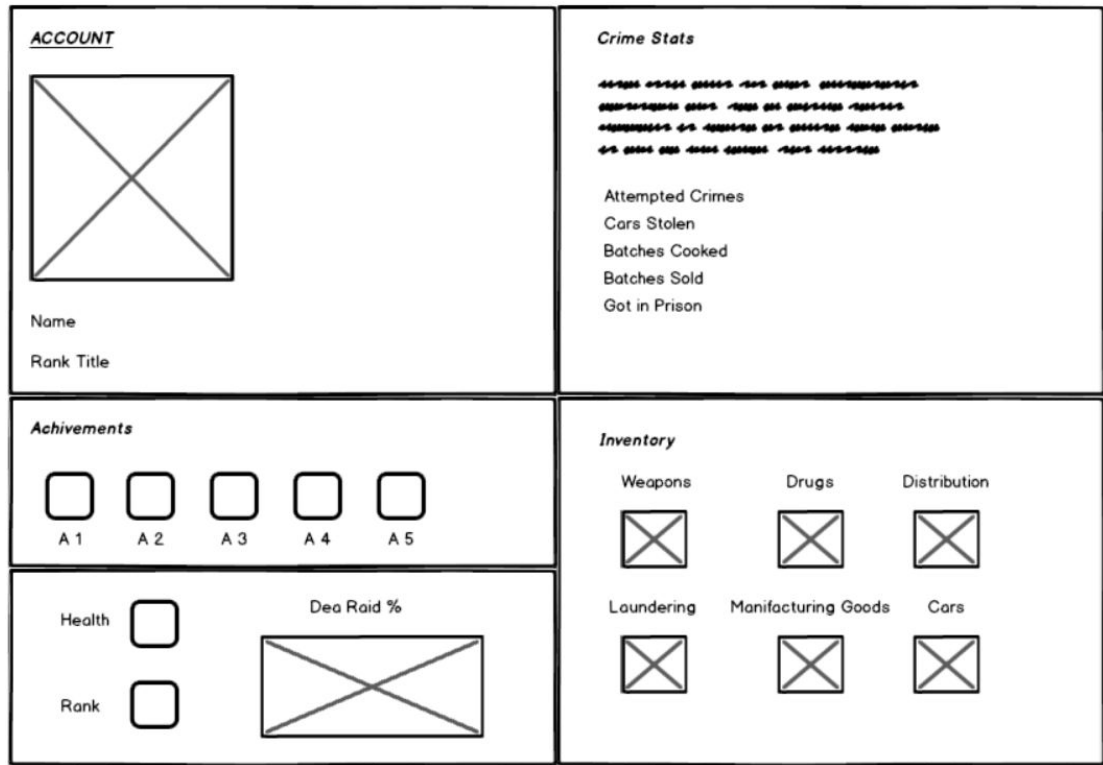
Figure-10: Profile Screen

Profile Screen will show everything that the player owns and has achieved previously. It will show the profile picture, name and title on the first dimension. Stats will be shown on the right side such as total batches sold, how many times the player got in prison. Below, there will be an Achivements screen that shows the fulfilled achievements. Inventory screen will show the player's tools for money laundering, drug manufacturing and owned cars. Lastly, there will be a screen to show the player's current health status, rank and chances for getting a DEA Raid.
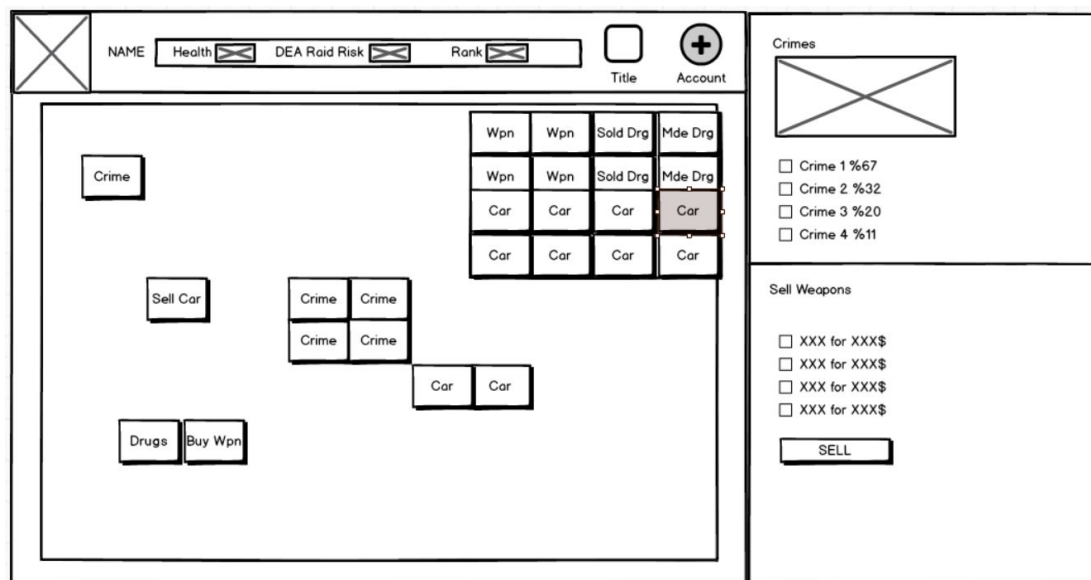
Figure-11: Main Game Screen

Main Game Screen will show the profile picture, name, health, rank and DEA Raid Chance on top briefly. It will also show the current title with an icon. User can open Account Screen pressing the button at the top-right.

Below the top box, the 9x9 map will be shown to the user that consists of variety of options such as selling and buying cars, committing crimes etc. the 4x4 at the top-right corner of the map will be reserved to the player to show player's tools, cars. Also it will show the drugs that are currently being manufactured and drugs that are currently being sold.

A submenu will appear on the right side of the Main Game Screen for every option for the user such as committing crimes, buying and selling cars and weapon dealing. In this example, we have shown the Crime submenu with an image on the top related to the crimes and crime options with their success percentages.
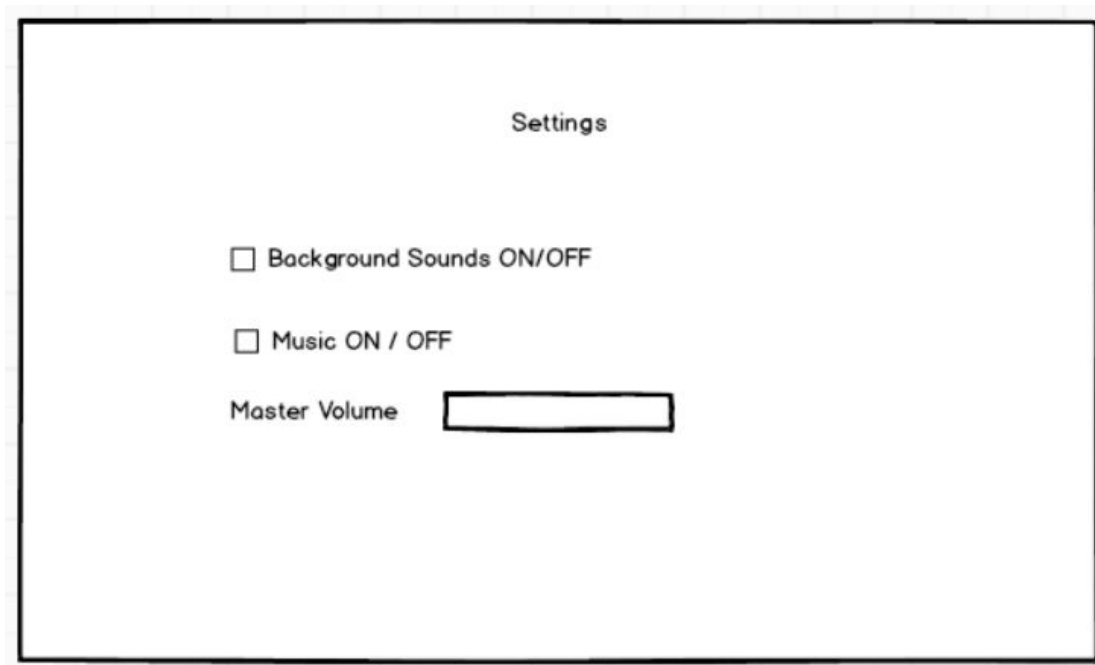
Figure-12: Settings Screen

Settings Screen will be a simple screen with 2 radio buttons for turning sounds and music on and off. There will be a horizontal bar for adjusting the Master Volume.
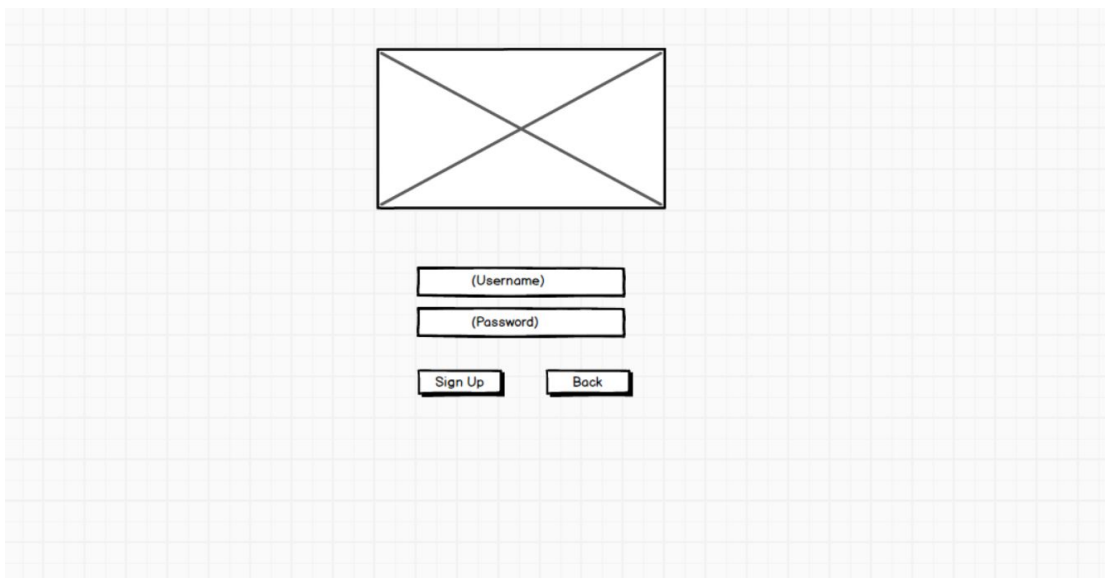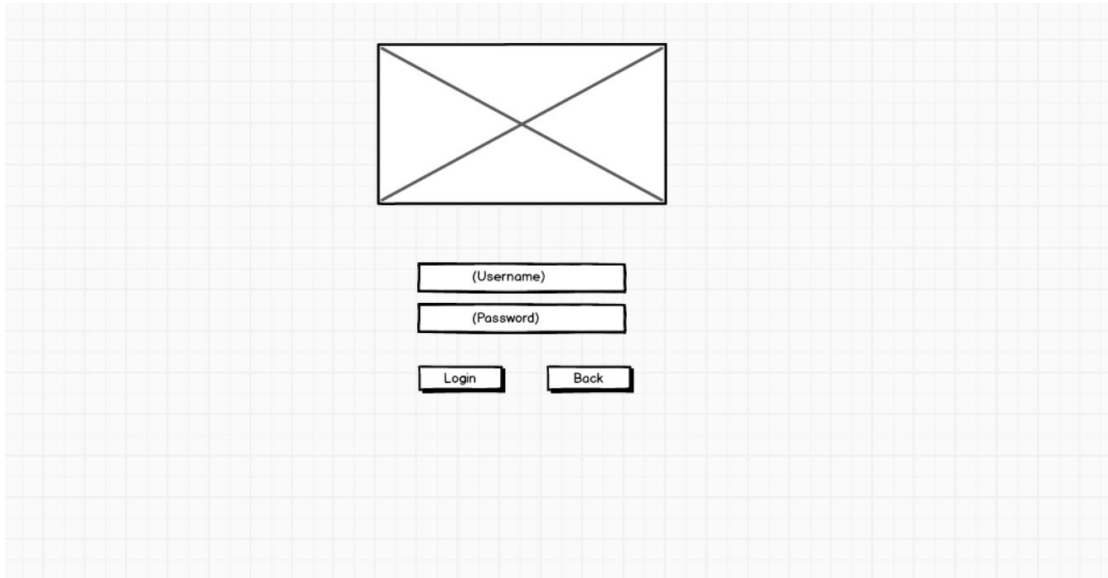

Figure-13: Sign Up Screen

Figure-14: Login Screen

Login and Sign Up screens will be similar screens with the game's image on top, Username and Password entry bars below. There will be a Login or Sign Up buttons on bottom left and Back button to return to Main Menu Screen.

References

- **[1], [2]** *Omerta*. **N.p., n.d. Web. 15 Feb. 2017.**

  <http://www.barafranca.gen.tr/>.
- *Clicking Bad*. **N.p., n.d. Web. 15 Feb. 2017.**

  <http://clickingbad.nullism.com>.