Bilkent University

Department of Computer Science

Senior Design Project

KEBAP TYCOON

[www.kebaptycoon.com]

Final Report

Group Members

Can Akgün
Doğancan Demirtaş
Kıvanç Kamay
Sinem Sav
Ulaş Sert

Supervisor

Hakan Ferhatosmanoğlu

Jury Members

Uğur Güdükbay
Selim Aksoy

Innovation Expert
Armağan Yavuz

Table of Contents

Table of Figures

Abstract

Our senior design project is a cross platform application called "Kebap Tycoon" which is a challenging game where players can found their restaurant chains and put effort to be the best in this sector by making decisions on both preferences of sales and locality strategies. The target user group will mostly depend on residences of Turkey as the game will represent instantaneous aspects of Turkish culture.

Keywords: Game, mobile, desktop, tycoon, Turkey, Turkish culture, multiplayer

1. Introduction

One of the contemporary sectors included in computer science field is surely entertainment sector. Today, the game industry has a big portion in economics; it employs dozens of people for different positions starting from game designer, game developer and all the way to the marketing executive. In this big market, business simulation games which are known as tycoon games became one of the most popular genres.

Businesses usually have tight set of rules and revolve around scarce resource allocation. Thus, in a way, maintaining a business in real world already is a sort of game. Consequently, if one wants to play that game, it's actually much easier to just turn on an electronic device and play one of the many great business tycoon simulations.

Within this genre, we decided to build a Turkish food business game, called Kebap Tycoon. This game will represent humorous sides of Turkish food and restaurant culture in which Turkish people can adore as well as where players can found their restaurant chains and put effort to be the best in this sector by making decisions on both preferences of sales and locality strategies.

In this report, we have written the finalized components of the senior project, engineering solutions and tools that are used. Moreover, at the end, we have included a user manual that will help increasing the user experience of the game.

2. Purpose of the Project

Its purpose is basically to entertain the players who chose to interact with our game. Moreover, the game itself embodies a purpose of adding a new game that is based on Turkish culture to the global game industry as well.

To achieve these purposes, we have utilized several main concepts. First of all, since our game targets Turkish people, humorous sides of Turkish food and restaurant culture are added to increase the entertainment. Moreover, we decided to add a multiplayer section where gamers can login with their Facebook accounts and see the situation of their friends' restaurants. Furthermore, the game will be implemented both for desktop and mobile platforms (Android and iOS) since the new era of games mainly revolves around these platforms.

3 Final Architecture and Design

In this section, the final architecture of Kebap Tycoon will be explained from the perspective of subsystem decomposition, object modeling, hardware/software mapping and persistent data management.

3.1 Package Diagram

Because of our LibGDX Game Engine, we needed to use the MVC architecture, which is applied to all components of the game such as customer, venue, staff and interface elements. Each component in the game contains a Model together with the Controller which updates the component upon either player or in-game action. The View is the display of the component. MVC structure is easy to apply on LibGDX, as LibGDX community highly recommends on building structure of projects with MVC architecture.

As our project has a requirement of holding the login credentials of the players and game state information for the saving/loading mechanism on a remote database service, we decompose the game into client and server layers.

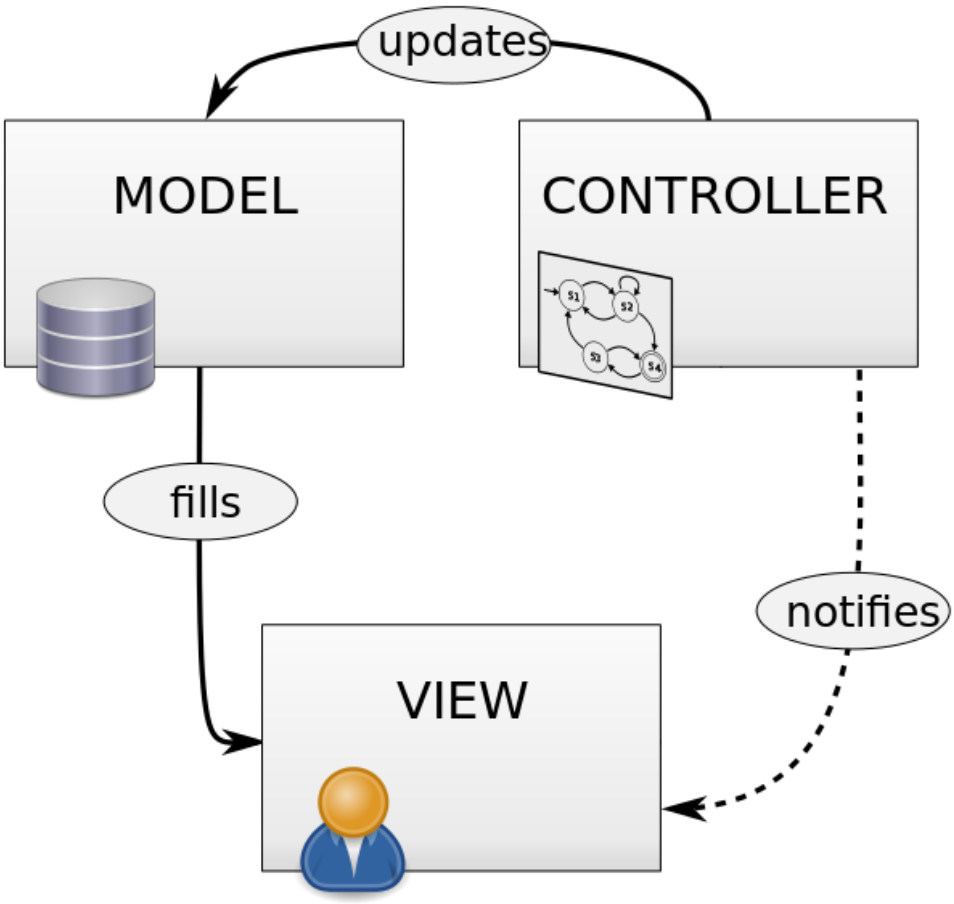
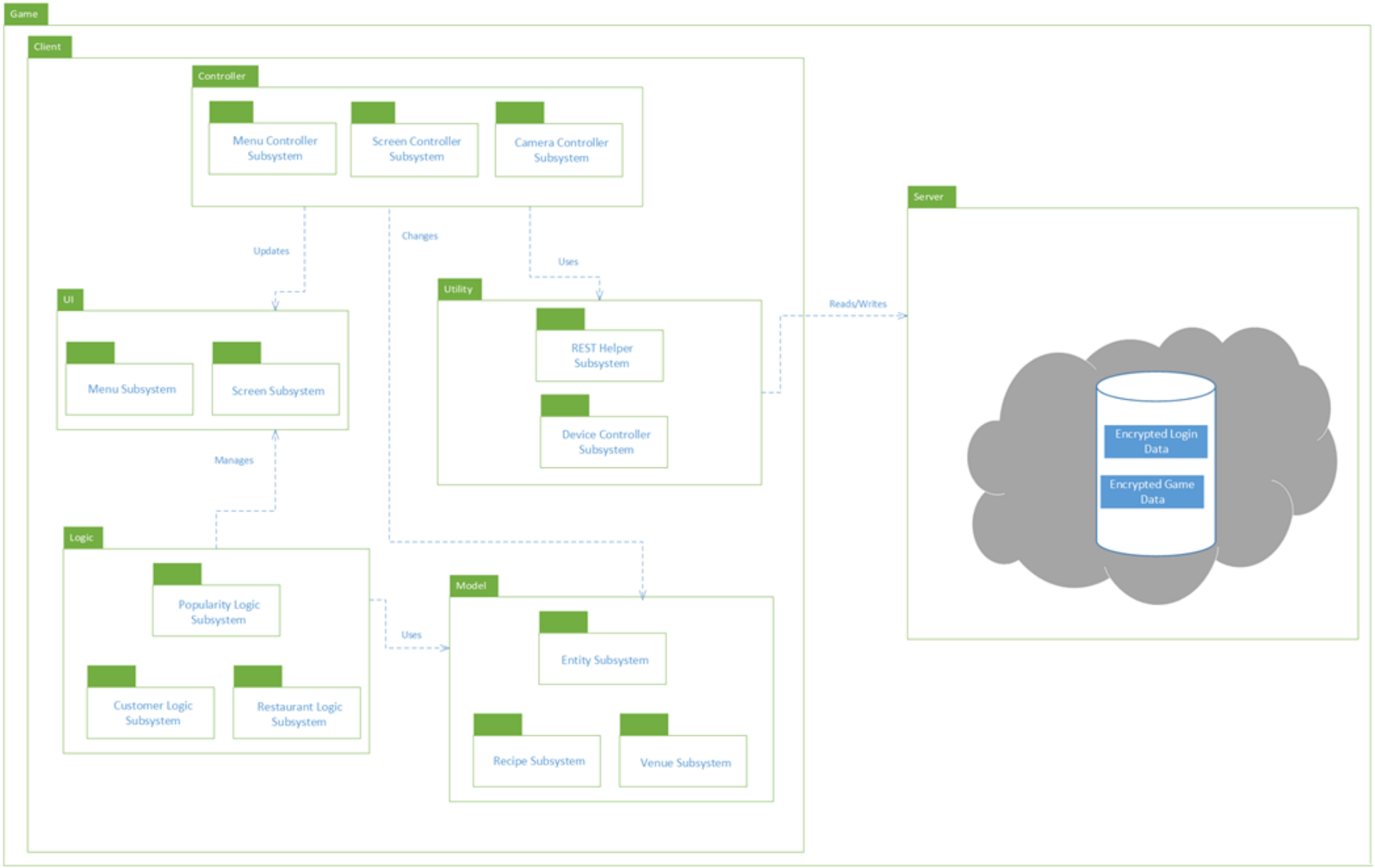


Figure 1 - Architecture of Kebap Tycoon Client Layer

3.2 Subsystem Decomposition

Subsystems of the game are created based on two major architectural elements which are Client and Server. With the help of Utility Subsystems these two major elements communicate between each other. The following figure is the graphical representation of the subsystem decomposition:


Game
Client
Controller
Menu Controller Subsystem
Screen Controller Subsystem
Camera Controller Subsystem
Updates
Changes
Uses
UI
Menu Subsystem
Screen Subsystem
Utility
REST Helper Subsystem
Device Controller Subsystem
Reads/Writes
Server
Encrypted Login Data
Encrypted Game Data
Manages
Logic
Popularity Logic Subsystem
Customer Logic Subsystem
Restaurant Logic Subsystem
Uses
Model
Entity Subsystem
Recipe Subsystem
Venue Subsystem

3.2.1 Client

Client layer uses the MVC architecture as described above, but additional components for connecting to remote database are required.

Controller Subsystems

Menu Controller Subsystem:

Responsible for managing the inputs that are coming from any of the menu screens while the game is being played. Each of the menu controllers are created according to the player's choices from the menu options. Thus, since only one controller can be active at a time, the remaining ones are pushed into the controller stack.

Screen Controller Subsystem:

All the game screens such as login screen, register screen, splash screen are handled by screen controller. The main responsibility of a specific screen's controller is managing the initiation of the menus that are covered by this screen.

Camera Controller Subsystem:

The camera controller manages the specific camera angle and the height (z-axis) that is most suitable for a tycoon game scene.

Model Subsystems

Entity Subsystem:

Responsible for storage of entities. An entity is any object within the system that can be rendered on the screen; such as customers, tables, chairs, etc. All properties and methods of entities are contained within here.

Recipe Subsystem:

Responsible for storage of customized recipes for dishes. A recipe of a dish can be modified by the player and customers will react differently to different recipes, such as making a recipe too spicy can cause them to complain.

Venue Subsystem:

Responsible for storage of all of the venue data inside the system; such as popularity, current stock of raw materials, employees etc.

Logic Subsystems

Popularity Logic Subsystem:

Manages the current popularity of any venue in the game based on the feedback customers give. Resulting popularity is viewed indirectly through the determination of the number of customers arriving.

Customer Logic Subsystem:

Manages the generation of new customers and customers themselves as; pathfinding, desired food, expectations, giving feedback, etc. The view of customers and their actions are lead through the UI over game logic.

Employee Logic Subsystem:

Manages the employees as; pathfinding, responsibilities, action queue, etc. The view of employees and their actions are lead through the UI over game logic.

UI Subsystems

Menu Subsystem:

The Menu Subsystem is responsible for showing the menus to the user and creation of their controllers.

Screen Subsystem:

The Screen Subsystem is responsible for showing the game screens to the user and creation of their controllers.

Utility Subsystems

Device Controller Subsystem:

The game uses device features such as Internet connection, media player and apps installed on device (Facebook). Also device state, screen orientation and language need to be retrieved by the game. That's why a device controller subsystem will be created to get native or physical device information and to make changes.

REST Helper Subsystem:

Kebap Tycoon REST API will be used in order to make information exchange between client and server. For ease of use, first a generic REST Helper subsystem will be created and then it will be used.

3.2.2 Server

We are using our own backend implementation on a Digital Ocean Ubuntu 14.04 server. The server has 20 GB SSD, 512 MB RAM and provides 1 TB traffic. Both the backend code and database will work on the same server.

3.3 Hardware – Software Mapping

Kebap Tycoon mainly targets mobile device users. In order to be played in Android devices, a Gradle build must be done and the “apk” file should be deployed to the device.

For iOS devices, the Xcode project should be built and linked with a developer account. Then the artifact must be deployed to the device (An Apple developer account is needed in that case).

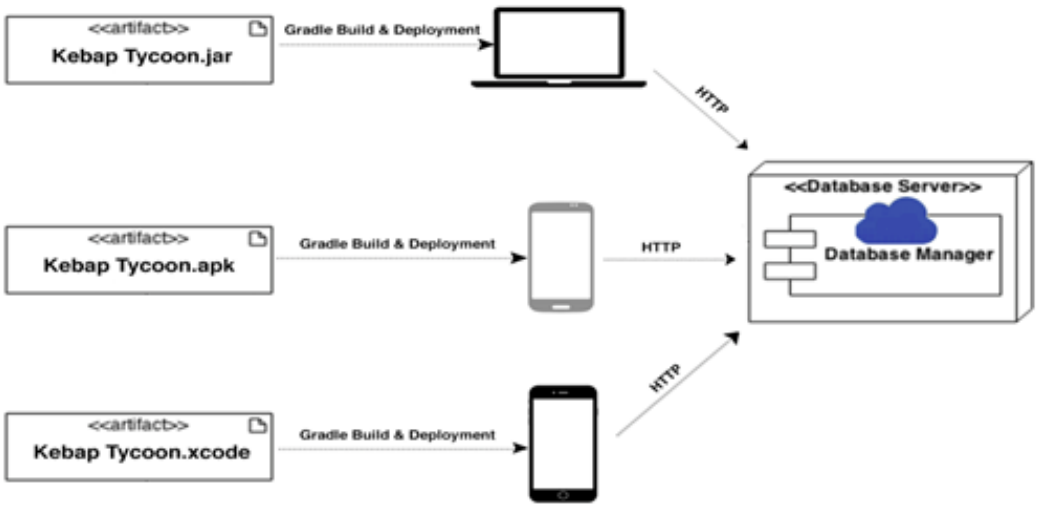
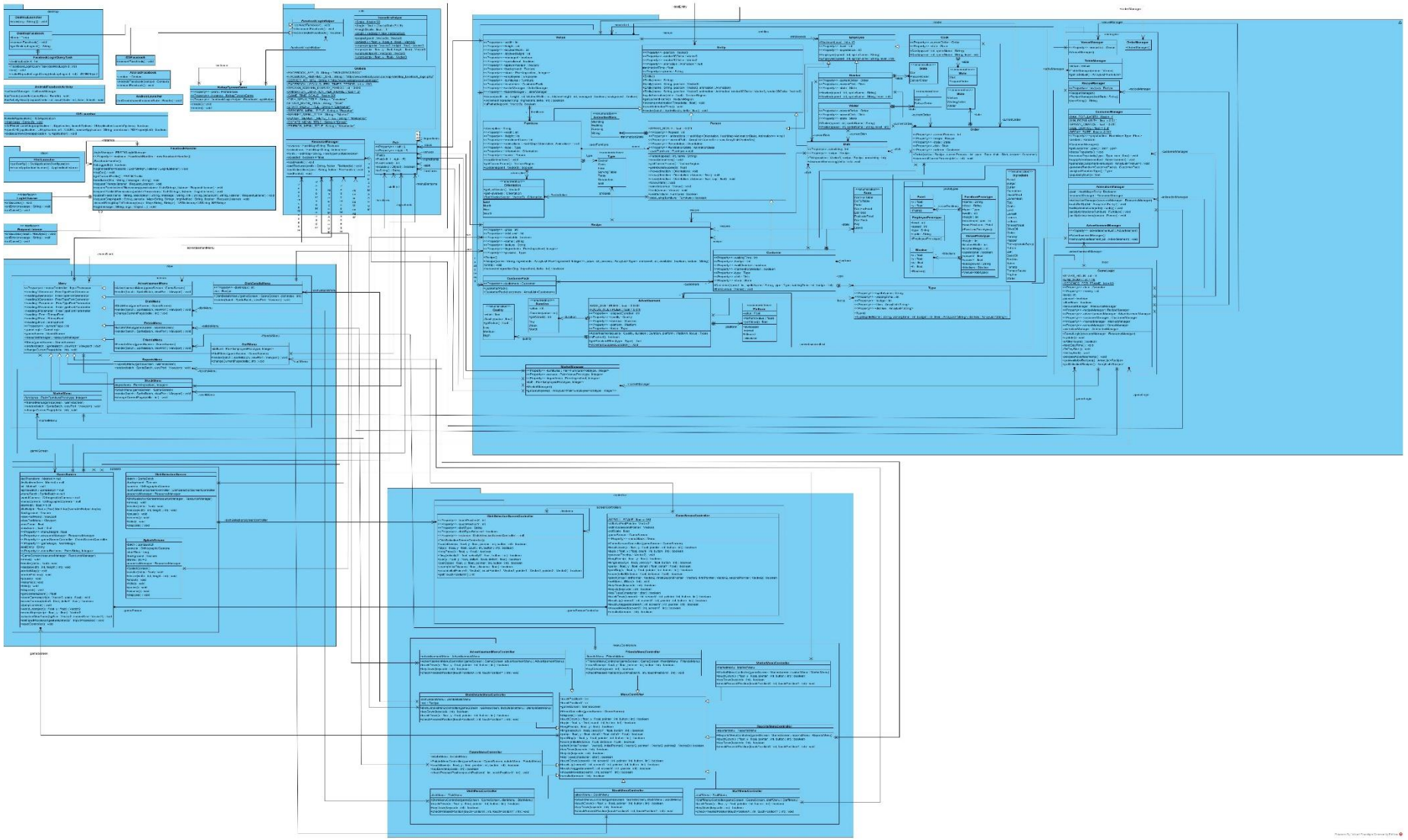


Figure 2 - Hardware-Software Mapping

3.4 Class Diagram



3.5 Persistent Data Management

In our system, we will use our own API to provide mechanisms for backend services that include managing and storing data. We will use database for storing a player's data i.e. amount of money, number of friends, item stocks, number of restaurants etc. Moreover, in our system, the players will not have the opportunity to save manually, rather than this, we have agreed to provide auto-save function. With this approach, we guarantee to have a persistent system and permanent data. However, if the system confronts errors due to the frameworks or services that we use (LibGDX), the current game progress can likely be interrupted and lost.

4. Impacts

4.1 Global Impacts

Technological tools, applications and games are becoming more and more customized. Kebap Tycoon will be under the field of tycoon games and appeal to Turkish people locally. Thus, if the game will be successful in the market place, it can set an example to other countries or we may try to develop different tycoon games which will address different countries' culture.

4.2 Economic Impacts

Contrary to many tycoon games, Kebap Tycoon will be free as a mobile application on application stores and as a desktop application. Thus, anyone preferring tycoon games will benefit from Kebap Tycoon. While it is beneficial for individuals' budget, Kebap Tycoon will also contribute to the global economy with commercials and in-app purchases. As a result, any individual who does not prefer paying for games will be able to get this game for free and ignore in-app purchases, while companies and Kebap Tycoon team will benefit from commercials.

4.3 Social Impacts

Kebap Tycoon, unlike other tycoon games, will support multi-player in a sense by giving an opportunity to see others' status on game. Through competition, the game will socialize people and change the way they play traditional tycoon games.

5. Tools and Technologies Used

REST API: Representational State Transfer that enables us to interact with our backend service (from anything that can send an HTTP request) [1].

LibGDX: Game-development application framework written in the Java programming language. It allows for the development of desktop and mobile games by using the same code base [2].

Android Studio: The Integrated Development Environment (IDE) that we used during the development process.

SourceTree: A free Git and Mercurial client for Windows and Mac [3].

RoboVM: RoboVM allows us to create native iOS applications using the same UI controls we would in Objective-C and Xcode, except with the flexibility and elegance of a modern language (Java), the power of the JVM library ecosystem, and a first class IDE - RoboVM Studio - at our fingertips [4].

6. Usage of Resources

During the development process, we mostly rely on the libGDX's official site since usage of a game engine is the core of our project. Moreover, for backend services that we are in need of, tutorials for both libGDX and RoboVM are examined to get deeper perspectives on these services.

In addition to these, question-and-answer websites for computer science such as "www.stackoverflow.com", helped us to get quick responses for our problems.

7. Software User Manual

7.1 Installation of the Game

7.1.1 Mobile Platforms

Mobile version of Kebap Tycoon will be specific to both Android and iOS platforms. Because of this reason, users of the platforms that want to play the game must use their own stores. Namely, Android users can download the game from Google Play and iOS users can download the game from AppStore.

7.1.2 Desktop

On the other hand, the desktop version will be available on our official site that is, “www.kebaptycoon.com”. Moreover, we think that putting the game on Steam can increase the amount of users. Hence, we have an idea of having the desktop version of the game on that platform as well.

7.2 Tutorial

- When the game is opened, players will login with their Facebook accounts.
- The game will provide an interactive tutorial at the very beginning of the game in which players can see different features. Players will be able to see their stock status and buy new materials necessary for the production of kebabs and maintenance of the restaurant.
- Players will be able to create new recipes and determine the prices of the kebabs and other products.
- Players will be able to hire and fire different qualified employees (chef, waiter etc.) necessary for their restaurant.
- Players will be able to see which kebab or products customers liked the most.
- Players will be able to see the restaurant’s and the dish’s popularity.
- Players will be able to buy different types of restaurant items and design the restaurant.
- Players will be able to see monthly reports about their restaurants.

- Players will be able to increase their restaurant's popularity with their recipes and items they buy.
- Players will be able to add their friends by using their nicknames or using Facebook.
- Players will be able to see their added friends' status on game including their profit, current restaurants etc.
- Players will be able to experience random good events where they gain money from.
- Players will be able to experience random bad events where they lose money. (ex. flood, robbery, fire).
- Players will be able to unlock items depending on their levels which are calculated by experience points.
- Players will be able to advertise their restaurant which will increase their revenue.
- Players will be able to open as many restaurants as they wish in case they have enough money.

References

[1] Parse.com, "Parse", 2015. [Online]. Available: https://parse.com/products. [Accessed: 19 Dec 2015].

[2] Libgdx.badlogicgames.com, "libgdx", 2015. [Online]. Available: http://libgdx.badlogicgames.com. [Accessed: 21 Dec 2015].

[3] Sourcetreeapp.com, “SourceTree”, 2016. [Online]. Available: https://www.sourcetreeapp.com/. [Accessed: 20.04.2016].

[4] Docs.robovm.com, “RoboVM”, 2016. [Online]. Available: http://docs.robovm.com/getting-started/introduction.html. [Accessed: 20.04.2016].