­­­­­­­­­­­­­

**Table Of Contents**

[1. INTRODUCTION 4](#_Toc350265124)

[1.1 Purpose 4](#_Toc350265125)

[1.2 Scope 4](#_Toc350265126)

[1.3 Glossary 4](#_Toc350265127)

[1.4 References 4](#_Toc350265128)

[2. OVERALL DESCRIPTION 5](#_Toc350265129)

[2.1 Product Features 5](#_Toc350265130)

[**2.1.1 Piccolo smartphone app main features (for the end user)** 5](#_Toc350265131)

[**2.1.2 Piccolo web-based management system (for the business owner)** 5](#_Toc350265132)

[2.2 User Characteristics 5](#_Toc350265133)

[2.3 General Constraints 5](#_Toc350265134)

[2.4 System Components 6](#_Toc350265135)

[2.5 System Architecture 6](#_Toc350265136)

[3. SOFTWARE REQUIREMENTS 7](#_Toc350265137)

[3.1 Software General Requirements – Shopper’s Application 7](#_Toc350265138)

[**3.1.1 Signup \ Login with Facebook** 7](#_Toc350265139)

[**3.1.2 Search for a shopping club** 7](#_Toc350265140)

[**3.1.3 Join a shopping club** 7](#_Toc350265141)

[**3.1.4 Browse shopping clubs** 7](#_Toc350265142)

[**3.1.5 Browse shopping club's offers** 7](#_Toc350265143)

[3.2 Software General Requirements – Management System 8](#_Toc350265144)

[**3.2.1 Signup \ Login with Facebook** 8](#_Toc350265145)

[**3.2.2 Create a new shopping club.** 8](#_Toc350265146)

[**3.2.3 Edit shopping club details** 8](#_Toc350265147)

[**3.2.4 Add a new offer for shoppers** 8](#_Toc350265148)

[**3.2.5 Remove an offer** 8](#_Toc350265149)

[**3.2.6 Update an offer** 8](#_Toc350265150)

[**3.2.7 View statistics of the club** 8](#_Toc350265151)

[**3.2.8 System landing page** 8](#_Toc350265152)

[3.3 Software Specific Requirements – Shopper’s Application 9](#_Toc350265153)

[**3.3.1 Functional Requirements** 9](#_Toc350265154)

[**3.3.1.1 Signup \ Login with Facebook** 9](#_Toc350265155)

[**3.3.1.2 Search for a shopping club** 12](#_Toc350265156)

[**3.3.1.3 Join a shopping club** 15](#_Toc350265157)

[**3.3.1.4 Browse shopping clubs** 18](#_Toc350265158)

[**3.3.1.5 Browse shopping club's offers** 21](#_Toc350265159)

[**3.3.2 Non-Functional Requirements** 24](#_Toc350265160)

[**3.3.3 Logical Database Requirements** 25](#_Toc350265161)

[**3.3.4 External Interface Requirements** 26](#_Toc350265162)

[**3.3.1.1 User Interface** 26](#_Toc350265163)

[**3.3.1.2 Hardware Interfaces** 26](#_Toc350265164)

[**3.3.1.3 Software Interfaces** 26](#_Toc350265165)

[**3.3.1.4 Communication Interfaces** 26](#_Toc350265166)

[3.4 Software Specific Requirements – Management System 27](#_Toc350265167)

[**3.4.1 Functional Requirements** 27](#_Toc350265168)

[**3.4.1.1 Signup \ Login with Facebook** 27](#_Toc350265169)

[**3.4.1.2 Create a new shopping club** 30](#_Toc350265170)

[**3.4.1.3 Edit shopping club details** 32](#_Toc350265171)

[**3.4.1.4 Add a new offer for shoppers** 34](#_Toc350265172)

[**3.4.1.5 Remove an offer** 36](#_Toc350265173)

[**3.4.1.6 Update an offer** 38](#_Toc350265174)

[**3.4.1.7 Statistics of the club** 40](#_Toc350265175)

[**3.4.1.8 System landing page** 42](#_Toc350265176)

[**3.4.2 Non-Functional Requirements** 44](#_Toc350265177)

[**3.4.3 Logical Database Requirements** 45](#_Toc350265178)

[**3.4.4 External Interface Requirements** 46](#_Toc350265179)

[**3.4.4.1 User Interface** 46](#_Toc350265180)

[**3.4.4.2 Hardware Interfaces** 46](#_Toc350265181)

[**3.4.4.3 Software Interfaces** 46](#_Toc350265182)

[**3.4.4.4 Communication Interfaces** 46](#_Toc350265183)

**1. INTRODUCTION**

**1.1 Purpose**

The purpose of this document is to specify the software requirements of the "Tetris-AI" project, and to overview the main components and functionality that shall be available in the first version of the application and environment.

**1.2 Scope**

The "Tetris-AI" is a project designed to provide an accessible and reliable API (application programming interface) for programmers who intend to write an AI (Artificial Intelligence) algorithm to the "classic-Tetris" game.

The API will provide functionality for reading game relevant information and sending key press commands to the Tetris game.

**1.3 Glossary**

|  |  |
| --- | --- |
| **Term** | **Definition** |
| API | Application Programming Interface |
| AI | Artificial intelligence – a computer program that simulates a human player of the Tetris game |
| Tetris | The classic Tetris game. |
| Emulator | A Nintendo based game emulator written in Java. The Emulator can run the Tetris game. |
| MSC | Message Sequence Chart |
| DB | Database |

**1.4 References**

**2. OVERALL DESCRIPTION**

**2.1 User Characteristics**

The end user is any programmer which is familiar with the Python programming language and the Tetris game and is aiming at writing an AI software that will play the game.

**2.2 Product Features/functionality**

The main component of the project is our Python library that enables a programmer to interface with the original Tetris game.  
The product will allow the user to:

* Receive information regarding the current game state (detailed in section..) from the game. Examples for game state information are the current score and the current tetrimino piece that is being played. (detailed in section..)
* Extract further data from the current game state. These data are calculated from the current game state and can be used as further input for the AI system.
* For example, the variance of the column height of the Tetris board and number of “holes” in the Tetris board. . (detailed in section..)

?

The project is split into 2 products, a smartphone app for the end users and a management system for the business owners.

**2.1.1 Piccolo smartphone app main features (for the end user)**

- Sign up with Facebook account

- Join or leave a shopping club

- Browse offers of a certain club (must be signed to this club)

**2.1.2 Piccolo web-based management system (for the business owner)**

- Create a new shopping club

- Manage club options (names, logos, photos)

- Add / Remove / Update new offers and benefits for shoppers

- View statistics of the club

All data of both the smartphone app and the management system will be stored on a designated database server. Therefore, an active internet connection will be required.

**2.3 General Constraints**

**Time limitation** - The first version of the application should be released before ?.

**2.4 System Components**

**2.4.1 AI**

Artificial intelligence – a computer program that simulates a human player of the Tetris game.   
It receives info regarding the current state of the Tetris game and performs a key press to affect the game accordingly.

**2.4.2 Tetris**

The classic Tetris game.

**2.4.3 Emulator**

A Nintendo based game emulator written in Java. The Emulator can run the Tetris game.

**2.4.4 Python library**

The core library that provides game info to the Tetris-AI algorithm and enables it to send key press commands to affect the gameplay.

**2.4.5 Python server**

A network-based mediator between the Java Bridge (defined below) and the Python library.

**2.4.6 Java bridge**

The component that is responsible for transferring and collecting the relevant data from and to the

emulator.

**2.4.7 Data handler**

The component that is responsible for reading and writing from and to the database

**2.4.8 DB - Database**

The Database stores information regarding AI-agents and their performance in previous games.

**2.5 System Architecture**

**3. SOFTWARE REQUIREMENTS**

**3.1 Software General Requirements – Shopper’s Application**

**3.1.1 Signup \ Login with Facebook**

Create a shopper's account using data from user's Facebook profile.

Use Facebook API to retrieve data about the user and create the account.

**3.1.2 Search for a shopping club**

User will provide a search string and get a result page of clubs with details that match the search.

**3.1.3 Join a shopping club**

User will be able to join a shopping club that he found by either using the search option or by scanning the shopping club's QR code.

**3.1.4 Browse shopping clubs**

User will get a list of all his clubs and will be able to navigate to a specific club.

**3.1.5 Browse shopping club's offers**

User will be able to see all open offers of a specific club (user must be signed to that club).

**3.2 Software General Requirements – Management System**

**3.2.1 Signup \ Login with Facebook**

Create a business manager account using data from user's Facebook profile.

Use Facebook API to retrieve data about the user and create the account.

**3.2.2 Create a new shopping club.**

User will fill a form with the following details:

- Business name

- Business type (select from an LOV)

- Business description

- Address

- Logo (upload an image file)

When the user submits the form, the data will be validated. If validation succeeds a new shopping club will be created and the user will be provided a QR code containing a link to the shopping club's page that he can use to promote his shopping club.

**3.2.3 Edit shopping club details**

Update a form with the following details:

- Business name

- Business type (select from an LOV)

- Business description

- Address

- Logo (upload an image file)

**3.2.4 Add a new offer for shoppers**

Fill offer description and upload image files.

Once the offer is created the user can choose to send notifications to shoppers. The user will be able to choose from different kind of offers.

**3.2.5 Remove an offer**

User will choose an offer from the list and disable or delete it.

**3.2.6 Update an offer**

User will choose an offer from the list and update its details (description/images).

**3.2.7 View statistics of the club**

User will be able to view a list of people registered to the club, how many times an offer was viewed and admitted (in case of action required offer).

**3.2.8 System landing page**

The landing page will be a single HTML page with info about the system. It will include an option to login, download the shopper app and signup as a shop owner.

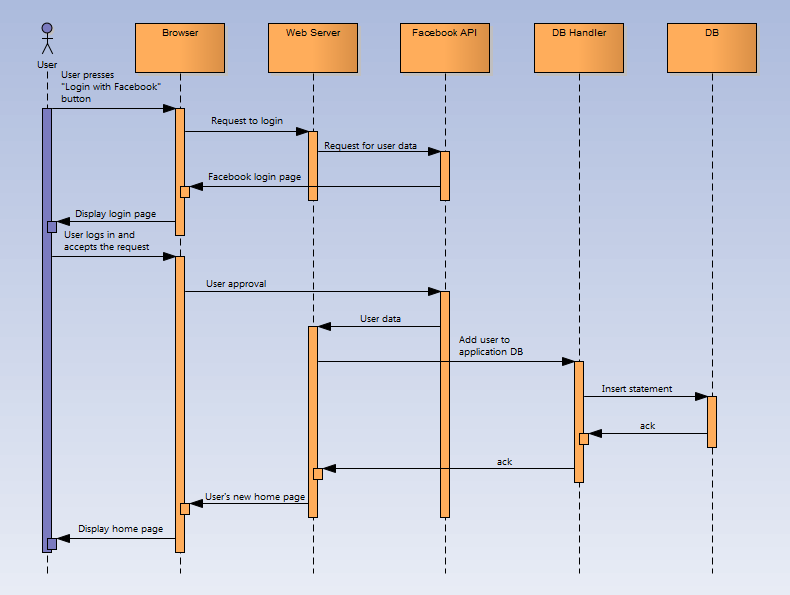
**3.3 Software Specific Requirements – Shopper’s Application**

**3.3.1 Functional Requirements**

**3.3.1.1 Signup \ Login with Facebook**

|  |  |
| --- | --- |
| **Priority** | Essential |
| **Trigger** | "Login With Facebook" button was pressed in the welcome screen |
| **Precondition** | The user is in the welcome screen |
| **Basic Path** | 1. The user presses "Login With Facebook" button from the welcome screen  2. If this is the 1st time, the Facebook app / browser takes focus (using the Facebook API) and asks the user to accept the permission required for the app to function.  3. After logged in with Facebook, the data will be stored on the DB. Data included the user profile and relevant data. If the operation failed, notify the user and stay in the welcome screen.  4. Login data is also saved to the local DB so user won't have to login again the next time he starts the app.  5. After successfully logging in, show the home page of the app.  6. If this was the 1st time a user logged in, show a welcome message. |
| **Post condition** | - The DB is updated with the user information (from Facebook)  - Login information is stored locally on the app  - User is taken to the home screen |
| **Exceptions** | - DB access errors (write / read)  - Local DB access errors (write / read) |

**MSC**

****

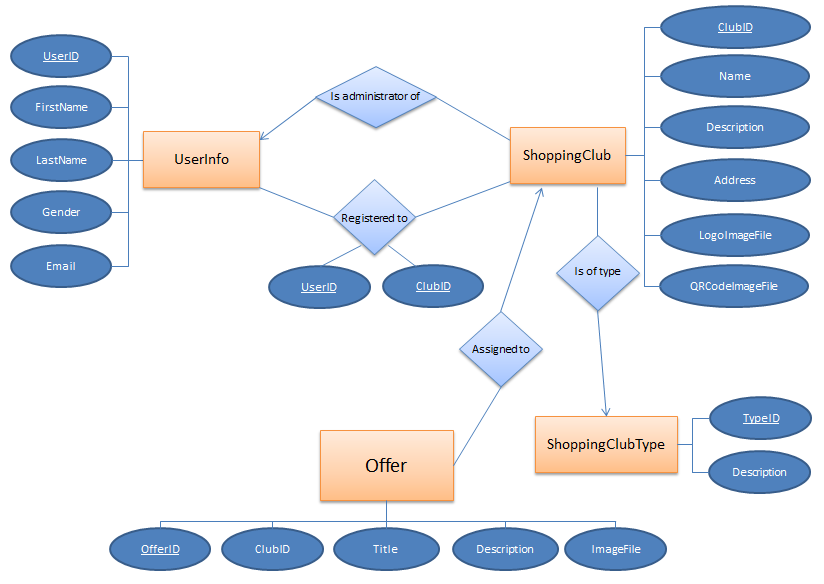
**Example Screen**



**3.4.2 Non-Functional Requirements**

* The application shall not change any of the operating system options or to any of the other applications that are installed. It shall not use or attempt to exploit any bugs and security issues with the user browser
* The application shall not access any data stored on the device except its own DB and data stored in the application own cookies.
* Response time won't exceed 5 seconds
* All the information that is asked to be deleted will be deleted from the application, including all Facebook credentials.
* The user interface will be easy to operate and will be simple so every user can operate the management system
* The application should work with all popular browser (Internet Explorer, Mozilla Firefox, Google Chrome, Opera, Apple Safari) in all the operating systems that support these browsers.

**3.4.3 Logical Database Requirements**

****

**3.4.4 External Interface Requirements**

**3.4.4.1 User Interface**

The UI shall be graphical.

The UI will be developed with HTML5 and JavaScript.

**3.4.4.2 Hardware Interfaces**

None.

**3.4.4.3 Software Interfaces**

The server side will be written in PHP programming language. The application shall interface with the following external software components:

**1. Apache HTTP Server** will be used as the application’s web server.

**2. MySql Database** will be used as the application’s database.

**3. Facebook** for managing registration and login.

**4. QR SERVER** for generating QR codes.

**3.4.4.4 Communication Interfaces**

None.