**FEATURE DESCRIPTION DOCUMENT**

* **Functional Requirements**
* **Feature name:** Registration

**Role(s):** This feature is used by every single person that at some point has the role of a customer.

**Feature description/steps:**

Registration is very important due to audience tracking and analysis. It enables a company to find the target clientele and adapt its offer to it. A user should enter info, for example his gender, age, email address, password and most importantly transaction number due to buying a ticket. After that, registration leads towards using the system. Unregistered customers cannot access the system.

**Dependencies/constraints:**

Email address should be valid. System checks that by sending 6-digit code to the mentioned address. Failing to validate the address 3 times, user is forbidden to continue. Password should be strong enough. System wants it to contain a number and also asks for repeating it twice.

* **Feature name:** Login

**Role(s):** Every registered customer.

**Feature description/steps:**

Login is the easier step. User should be able to open the page by entering a valid email address and password.

**Dependencies/constraints:** If password or email address is not entered correctly, you will not be able to open page with your user account.

* **Feature name:** Find a movie

**Role(s):** Customers that successfully passed login and intend to book something

**Feature description/steps:**

Scrolling down the list of the movies should be categorized and the “select” button should be visible. You will be able to see short description of a movie, main actors, duration of a movie and even small movie picture.

**Dependencies/constraints:** Customer won’t be able to pick any movie he wants at any time he wants. Finding a movie depends on repertoire.

* **Feature name:** Time

**Role(s):** Customers that picked a movie title

**Feature description/steps:**

Movies almost always have multiple screenings. Customer should pick the one that suits him.

**Dependencies/constraints:** If many people are interested in some movie, it will probably start later than it should have started.

* **Feature name:** Seat viewing

**Role(s):** Customers that picked movie title and suitable time

**Feature description/steps:**

A display of available places should appear at this moment. Customer chooses one or more which are marked with different color than those already booked.

**Dependencies/constraints:** If you come with a group of people (5-6 visitors), you will probably not be able to pick all seats one next to each other in the row if numerous of 2 booked seats have been made.

* **Feature name:** Generate ticket

**Role(s):** Customers finishing their reservation

**Feature description/steps:**

Users should receive some kind of affirmation ticket. System should be able to generate ticket within few seconds with code which will include (count) one more movie guest and decrease number of seats.

**Dependencies/constraints:** If system slows for some time and does not accept immediately some previous booking confirmation, and at the same time another user picks the same movie, ticket will not be generated.

* **Feature name:** Customers finishing their reservation

**Role(s):**

**Feature description/steps:** System should only allow users to move to payment only when mandatory fields above are checked and completed. The payment is mapped and checked to see if it should be stored for processing later, or submitted to the payment gateway. For payments that are stored, a scheduler task is created to check stored payments if they are ready for release. If so, the payments are submitted to the payment gateway.

**Dependencies/constraints:** (is this feature dependent on some other features or factors; are there some constraints for this feature)

* **Nonfunctional Requirements**
* **Examples (delete this section in your document)**

For your **functional requirements**, please use the above template, which is also demonstrated in the following few examples. You should have **at least 10** functional requirements, but do not limit yourselves with this number; feel free to add as many features as you need or want.

For the **nonfunctional requirements**, you can simply write them down as sentences. You should have **at least 5** nonfunctional requirements, but you are welcome to add more.

* **Functional Requirements**
* **Nonfunctional Requirements**

1. Security –

The system uses SSL (secured socket layer) in all transactions that include any confidential customer information. The system must automatically log out all customers after a period of inactivity. The system should not leave any cookies on the customer’s computer containing the user’s password. The system’s back-end servers shall only be accessible to authenticated administrators. Sensitive data will be encrypted before being sent over insecure connections like the internet.

2. Reliability –

The system provides storage of all databases on redundant computers with automatic switchover. The reliability of the overall program depends on the reliability of the separate components. The main pillar of the reliability of the system is the backup of the database which is continuously maintained and updated to reflect the most recent changes. Thus the overall stability of the system depends on the stability of container and its underlying operating system.

3. Availability – The system should be available at all times, meaning the user can access it using a web browser, only restricted by the downtime of the server on which the system runs. In case of an of a hardware failure or database corruption, a replacement page will be shown. Also in case of a hardware failure or database corruption, backups of the database should be retrieved from the server and saved by the administrator. Then the service will be restarted. It means 24 X 7 availability.

4. Maintainability – A commercial database is used for maintaining the database and the application server takes care of the site. In case of a failure, a re-initialization of the program will be done. Also, the software design is being done with modularity in mind so that maintainability can be done efficiently.

5. Portability – The application is HTML and scripting language based. So The end-user part is fully portable and any system using any web browser should be able to use the features of the system, including any hardware platform that is available or will be available in the future.