System Requirements and Design

Team Hackermen

TicketSalad

Team Members:
John Smith
Thato Mothusi
Jarryd Baillie
Brandon Texeira
Thomas Honiball
Tristan Joseph

Client:
Tribus Digita

Contents

1 System Overview

1.1 Purpose

The purpose of the TicketSalad application is to provide users with a platform which they are able to spend a minimal amount of money in order to stand a chance to win a ticket valued at a significant amount with ease.

1.2 Project Scope

The TicketSalad system is a lottery based ticket platform. A user registers their email and credentials, they then purchase credits. Using these credits, user's can 'bid' on events displayed to them. When the user bids on an event they are required to input a 6 digit number. If the 6 digit number they input is the same number that the application randomly generated, the user wins the ticket.

1.3 Definitions, acronyms and abbreviations

- UML - Unified Modeling Language
- CRUD - Create Update Delete
- Credits - Credits is a form of currency in the TicketSalad application used to bid on tickets. Credits are purchased using credit cards.

1.4 UML Domain Model

NEEDS TO BE DONE

2 Functional Requirements

2.1 Users

The TicketSalad application will have two users, namely a regular user and then an admin. The system should allow regular users to do the following:

- Register or log into the system.
- Edit/Update their info
- Purchase credits
- Bid for events

- View information on events.

The system should allow the admin user to do the following:

- Perform CRUD operations on all the available events.
- View statistics and data on all events

2.2 Sub Systems

The TicketSalad System can be broken into 2 subsystems, namely the application user subsystem and admin subsystem.

The application user subsystem can then be further broken down into 3 subsystems, the events subsubsystem, the user subsubsystem and the notification subsubsystem.

- Admin subsystem - This subsystem is responsible for admins being able to modify and add events.
- Application user subsystem - This subsystem is the application that regular users will use.
- User subsubsystem - This subsubsystem is responsible for the users details credentials, and credits.
- Events subsubsystem - This subsubsystem is responsible for all actions performed on/by events by regular users.
- Notification subsubsystem - This subsubsystem is responsible for creating and handling notifications.

UML COMPONENT DIAGRAM NEEDS TO BE DONE

2.3 Specific Requirements

Admin subsystem
R1.1 The admin portal must allow an admin to log in.
R1.2 The admin portal must allow an admin to log out.
R1.3 The admin portal must allow an admin to add an event.
R1.4 The admin portal must allow an admin to update an event.
R1.5 The admin portal must allow an admin to delete an event.
R1.6 The admin portal must allow an admin to view which user has won an event.

User subsubsystem
R2.1.1 The TicketSalad application must allow a user to register.
R2.1.2 The TicketSalad application must allow a user to log in.
R2.1.3 The TicketSalad application must allow a user to log out.
R2.1.4 The TicketSalad application must allow a user to edit their information.
R2.1.5 The TicketSalad application must allow a user to view their information.
R2.1.6 The TicketSalad application must allow a user to enter their card or online banking details.
R2.1.7 The TicketSalad application must allow a user to view how many credits they have remaining.
R2.1.8 The TicketSalad system must allow a user to purchase more credits.

Events subsubsystem
R2.2.1 The TicketSalad application must allow a user to view all of the current events.
R2.2.2 The TicketSalad application must allow a user to search for specific events.
R2.2.3 The TicketSalad application must allow a user to bid on an event.
R2.2.4 The TicketSalad application must decrease a users credits once they have made a bid on an event.

Notification subsubsystem
R2.3.1 The TicketSalad application must subscribe a user to an event once they have made a bid.
R2.3.2 The TicketSalad application must notify a user if they have won an event.
R2.3.3 The TicketSalad application must notify a user on all details relating to their subscribed events.
A ton of diagrams need to be done here

3 Non-functional requirements

- The system must respond to any input within 1 second.
- The users data must be kept secure at all times.
- The systems data and information must be available at all times.

4 System Architecture

4.1 Interfaces

4.1.1 User interfaces

- **Splash screen** - The splash screen is the first screen that the user is exposed to this screen contains the TicketSalad Logo and an 'about us' section. The user then has the option to sign in or sign up.
- **Sign Up screen** - The sign up screen contains the form that the user will use to enter their details in order to register an account for TicketSalad.
- **Sign In screen** - The sign in screen is where the user enters their details to sign into the TicketSalad System.
- **Home screen** - The home screen is the main screen of the application, here the user is exposed to all the available events and they are able to claim on these events directly from the home screen.
- **Profile Screen** - This is the screen that the user will use to view their profile, they can view their username, profile picture and remaining credits on this screen. The user can then select to buy more credits, edit their profile, go to the contact us screen, or log out.
- **Edit Profile Screen** - Here the user can edit all their personal details on the system.
- **Buy Credits screen** - This is the screen where the user is able to purchase more credits in order to bid on events.

4.1.2 Hardware interfaces

4.1.3 Software interfaces

4.2 Architectural styles

The entire TicketSalad system is based off a client-server model. However the client side of the system, uses an MVC model. The reason we have chosen these architectures is so that the data that will be presented to the user is stored on the client side thus making ensuring saftey and integrity of the data. With all the data stored on the client side this will allow only the admin to perform all CRUD operations on the data and still allow the

user to access and view the data, thus allowing them to see events claim on them and possibly win them.

4.3 System configuration

Refer to the System Configuration image in the user manual for a a deployment diagram on the TicketSalad System.

All of the system's data is stored on the client side on a mongoDb server, the server is then accessed and altered using a meteor web server, which is accessed by the user using a TCP/IP connection from their mobile device.