Software Design Specification

for

Melodious

**Version 1.8 approved**

**Prepared by musicLovers**

**Melody**

**05/17/2019**

**Table of Contents**

**Table of Contents ii**

**Revision History ii**

**1.** **Introduction 3**

1.1 Goals and objectives

1.2 Statement of scope

1.3 Software context

1.4 Major constraints

**2.** **Data design 3**

2.1 Internal software data structure

2.2 Global data structure

2.3 Temporary data structure

2.4 Database description

**3.** **Architectural and component-level design 4**

3.1 System Structure

3.2 Description for Component

**4.** **User interface design 5**

4.1 Description of the user interface

4.2 Interface design rules

4.3 Components available

4.4 UIDS description

**5.** **Restrictions, limitations, and constraints 6**

**6.** **Testing Issues 6**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Camy  Linh  Hailey | 04/24/2019 | Divide the document section among team members. | 1.0 |
| Camy  Linh | 04/29/2019 | Add ticket purchase UI  Draft database description, update software context 1.3 | 1.1 |
| Camy | 04/30/2019 | Add user dashboard UI | 1.2 |
| Camy | 05/07/2019 | Add registration and log-in test class and constraints. | 1.3 |
| Camy Linh Hailey | 05/08/2019 | Add use case and sequence diagrams | 1.4 |
| Linh | 05/09/2019 | Complete component Melodious Social Media  Complete statement of scope | 1.5 |
| Camy | 05/12/2019 | Complete component Melodious User Registration and update section 5 | 1.6 |
| Linh | 05/13/2019 | Update the introduction, major constraints 1.4, limitations on section 5 | 1.7 |
| Linh, Hailey, Camy | 05/17/2019 | Fix font size and spacing issues | 1.8 |

# Introduction

This document describes the software design specification of the Melodious website and application. It clarifies the project scope and goals, data design, architectural and component-level design, user interface design, system constraints, and testing issues.

## Goals and objectives

The goal for this project is to provide an interactive ticket selling system for concertgoers and to give uprising artist a platform to promote their events. Unlike Ticketmaster or StubHub that sells tickets for sport, theater, and concerts, the Melodious system focus only on local and international music events. Our final product will benefit uprising artists, users, and Melodious’ stakeholders.

The end-users will gain benefits from using the system. They will be able to search and buy tickets easily, save points and rewards for future events, and build connections with other users or artists. Uprising artists will be able to reach out to more people and promote themselves through our system. Since the system is free for all users, stakeholders from Melodious company will generate profits through commission and advertisements.

## Statement of scope

Melodious is an integrated system of ticket selling platform, points and rewards system, social media platform, and uprising artists promotion. The system also allows admin to upload concert videos in the past. It is a combination of Facebook, Ticketmaster, and Youtube. The system targets mainly to those who have strong interest in music. Major inputs include event lists from Stubhub and Ticketmaster and data from users. Outputs includes information about events, tickets, and payments. Additionally, the system responds depending on users’ actions such as posting contents or sharing posts. According to stakeholders’ requirements, here are the ranking for specific system requirements. The registration component and the ticket selling component are essential and must be developed and completed. The Melodious social media platform, uprising artists promotion and the rewards system are classified as the desirable requirements. Uploading concert videos is a future requirement because it requires advanced research in video resources and digital copyright.

**Major inputs**

* Event lists from Stubhub and Ticketmaster
* Data from users including first name, last name, date of birth, email address, password, payment information, shipment information, and social media posts

**Major outputs**

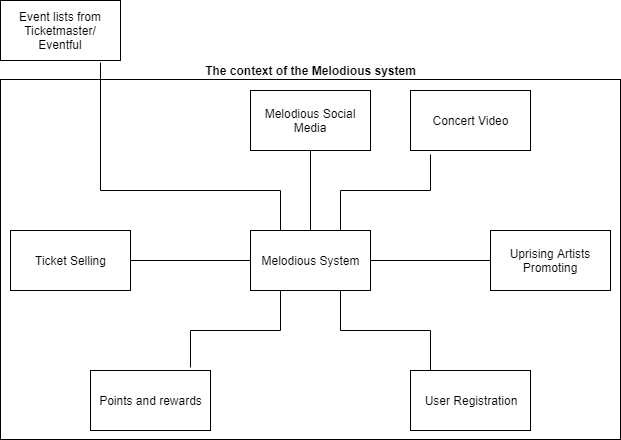
* Event ticket information generated after the user’s complete transactions
* Social media contents posted by users
* Actions from users

**Processing functionalities**

|  |  |
| --- | --- |
| **Component** | **Ranking** |
| **Registration Component** | Essential |
| **Ticket Selling Component** | Essential |
| **Melodious Social Media Platform** | Desirable |
| **Uprising Artist Promotion** | Desirable |
| **Rewards Component** | Desirable |
| **Uploading Concert Videos** | Future requirement |

## Software context

The Melodious website and mobile application are a new self-contained product of the Melody company. It provides users with a convenient and user-friendly environment so that they can stay up-to-date with current music events, purchase concert tickets, connect with uprising artists, and share their music interests. The system includes a ticket selling platform, a social media platform, points and rewards component, user registration system and video component. The system will receive event lists by integrating concert data and information from Ticketmaster and Eventful. Figure 3.1.a describe the context of Melodious system.



**Figure 3.1.a Context diagram for Melodious Registration**

## Major constraints

The user interface design shall adhere to the brand mark guidelines of the Melody music company to ensure the brand consistency of the organization.

# Data design

A description of all data structures including internal, global, and temporary data structures.

## Internal software data structure

Melodious will not incorporate any internal data structure. The database tables alone will suffice for the storage and retrieval of data. The data from the queries will be run and stored through traditional tables without any additional data segments.

## Global data structure

The information that will be available is for the user, system/application, and system features. The global data is read-only to users who aren’t part of the administrative group who will have read/write access to the database.

## Temporary data structure

There is no interim file needed for this system.

## Database description

The Melodious system includes databases that store information of regular users, artists, admins, and music events.

|  |  |  |  |
| --- | --- | --- | --- |
| **Regular User** | | | |
| **Column** | **Data Type** | **Constraint** | **Description** |
| UID | Int | Primary Key | Unique ID of a user |
| Ufname | Varchar (50) | Not null | User first name |
| Ulname | Varchar (50) | Not null | User last name |
| Uemail | Varchar (50) | Not null | User email address |
| Udob | Date | Not null | User date of birth |
| Uphone | Int | Not null | User phone number |

|  |  |  |  |
| --- | --- | --- | --- |
| **Artists** | | | |
| **Column** | **Data Type** | **Constraint** | **Description** |
| AID | Int | Primary Key | Unique ID of artist |
| Afname | Varchar (50) | Not null | Artist first name |
| Alname | Varchar (50) | Not null | Artist last name |
| Aemail | Varchar (50) | Not null | Artist email address |
| Adob | Date | Not null | Artist date of birth |
| Aphone | Int | Not null | Artist phone number |
| Snum | int | Not null | Number of songs |

|  |  |  |  |
| --- | --- | --- | --- |
| **Music events** | | | |
| **Column** | **Data Type** | **Constraint** | **Description** |
| MID | Int | Primary Key | ID of an event |
| Mname | Varchar (50) | Not null | Event name |
| Mdate | Date | Not null | Event date |
| Mtime | Time | Not null | Event time |
| Madd | Varchar (50) | Not null | Event address |

|  |  |  |  |
| --- | --- | --- | --- |
| **Ticket transaction** | | | |
| **Column** | **Data Type** | **Constraint** | **Description** |
| TID | Int | Primary Key | ID of a transaction |
| Tnum | Int | Not null | Numbers of tickets |
| Tdate | Date | Not null | Transaction date |
| Ttime | Time | Not null | Transaction time |
| Ptype? | Varchar (50) | Not null | Payment type |

# 

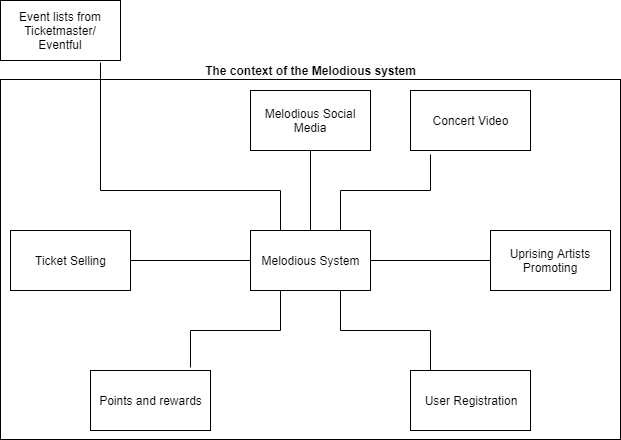
# Architectural and component-level design

## System Structure

## A detailed description the system structure chosen for the application is presented. Diagrams (and accompanying text) are a must for this description. You should show at least one structure diagram (e.g., Class diagrams) and, if helpful, one or more interaction diagram (e.g., sequence diagrams)

### Architecture diagram

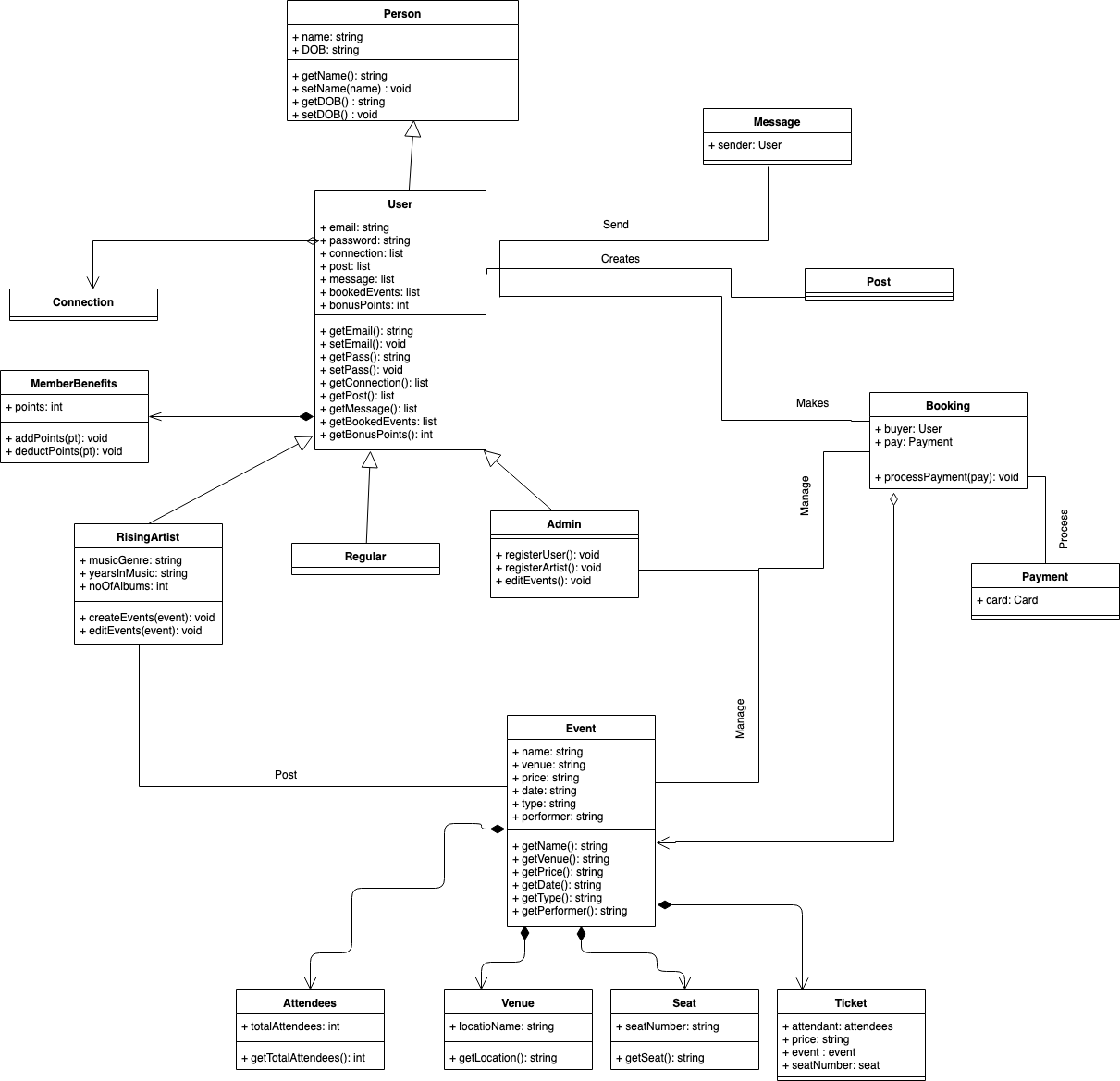
The Melodious system is a combination of a shared data repository architecture and a client-server architecture. It has object oriented decomposition and use event based for the execution style.



**Figure 3.1.a Context diagram for Melodious Registration**

### Class diagram

Below is the decomposition of entities in the Melodious application in to classes.

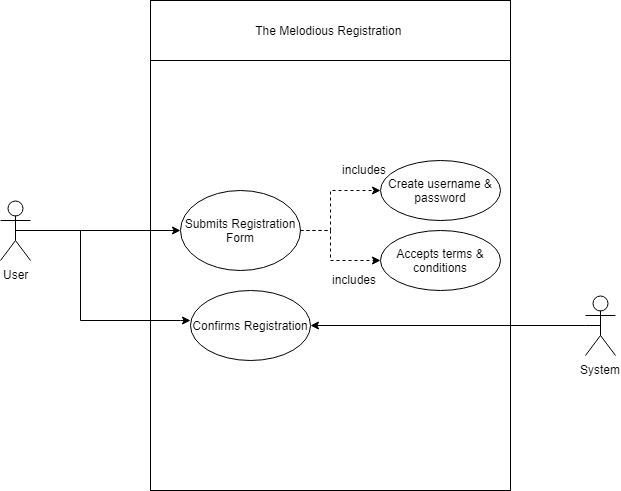


**Figure 3.1.b Class diagram for Melodious Registration**

## Description for Component User Registration

Author - Camy Chhetri

The user registration is the component of the Melodious application that will grant the user access to the system. Figure 3.2.a is a use case diagram describing different features of the Melodious Registration.



**Figure 3.2.a Use case diagram for Melodious Registration**

### Processing narrative for component Melodious User Registration

The registration component will have an interface. The user will register to the system by submitting the registration form such as first name, last name, date of birth, sex, and email. The component is responsible for sending email to the user and granting appropriate access to the user.

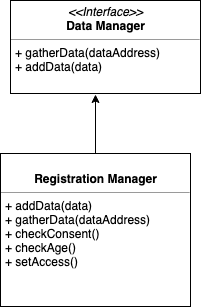
### Component Melodious User Registration interface description

The Melodious Registration component takes first name, last name, date of birth, sex, and email, sends them to the servers, and creates a user in the system if all the criteria are met.

### Component Melodious User Registration processing detail

The component receives input from users and performs age and email validation. A detailed algorithmic description for each component is presented.

#### Design Class hierarchy for component registration



**Figure 3.2.b Class Diagram for Melodious User Registration**

#### Restrictions/limitations for component Melodious User Registration

3.2.3.2.1 User must have a valid email address to register.

3.2.3.2.2 User must be at least 13 years old to register.

3.2.3.2.3 The interactions between the users and the system are in real time. Specifically, the system must response to the users’ actions within 3 seconds.

#### Performance issues for component Melodious User Registration

3.2.3.3.1 The user may have to wait 2 to 3 minutes to get confirmation registration link in their email.

#### Design constraints for component Melodious User Registration

3.2.3.4.1 The system is available in English only.

3.2.3.4.2 To register as an artist, a user must use different registration form which is called artist registration form and provide extra information to verify the artist.

#### Processing detail for each operation of component Melodious User Registration

##### 3.2.3.5.1 When the user clicks a ‘Register’ button, they are navigated to the User Registration Page.

3.2.3.5.2 The user fills out the sign-up form with their first name, last name, email, password, sex, and date of birth and checks ‘terms and conditions’ checkbox.

3.2.3.5.3 The user clicks ‘Sign Up’ button.

3.2.3.5.4 Their information is sent to the server.

3.2.3.5.5 The server validates their information.

3.2.3.5.6 The email verification is sent to user’s email.

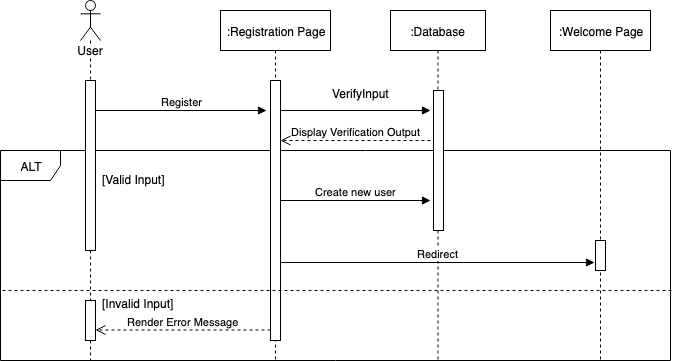
3.2.3.5.7 The user clicks the verification link.

3.2.3.5.8 The user is navigated to the dashboard page.

### Dynamic Behavior for Component Melodious User Registration

The user fires up the registration process by registering using the Registration Page class and the Register Page class performs verification by checking if the email already exists in the Database. The Registration Page class displays verification output. If the inputs were valid then the Registration Page class creates new user in the Database and redirects the user to the Welcome Page class. If the inputs were invalid, then the Registration Page class renders error message to the user.

### Interaction Diagrams



**Figure 3.2.c Sequence diagram of Melodious User Registration**

## Description for Component Ticket Purchasing

Author - Hailey Wilson

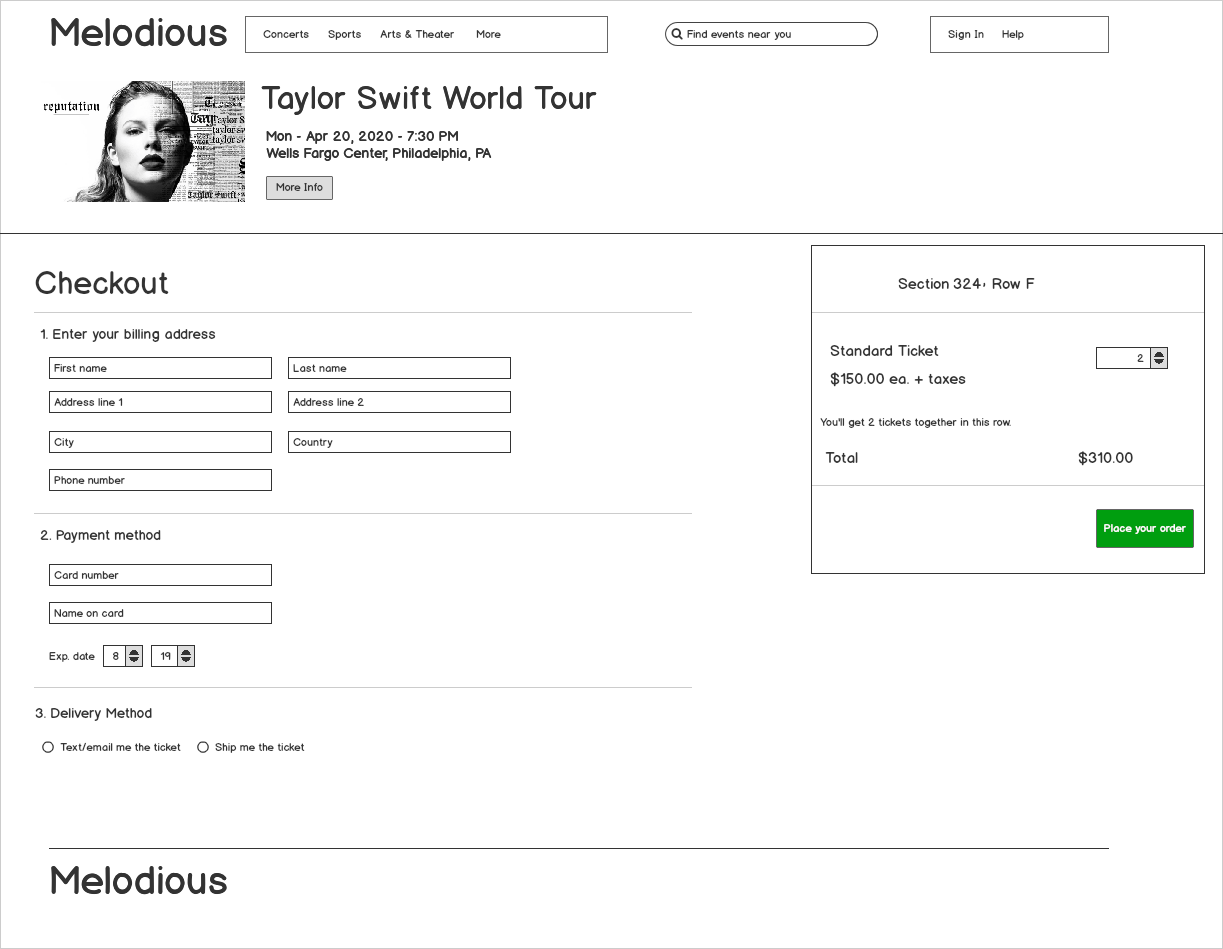
The ticket sales is the component of the Melodious application that will allow the user to purchase tickets on the system.

### Processing narrative for component ticket purchasing.

This section will contain a description of the responsibilities of the ticket sales component. The ticket sales component of Melodious has many responsibilities. The following component must allow registered users to be able to purchase up to five (5) tickets per event advertised on the Melodious website and app. It is also responsible for allowing registered users to enter their credit card information for payment and validate the payment to ensure it is a valid for of payment. The users also have an option to save or forget the credit card information they have entered into the system.

### Component ticket purchasing interface description.

There are two steps for purchasing tickets on Melodious. The first step is for the users to select an event, specify how many tickets they would like, and place the tickets in their cart for checkout. The user interface for the second step can be seen pictured below, Figure 3.3. For the user to complete their purchase they must enter their billing address, payment method, choose a delivery method, and then click the green button labeled “Place your order”.

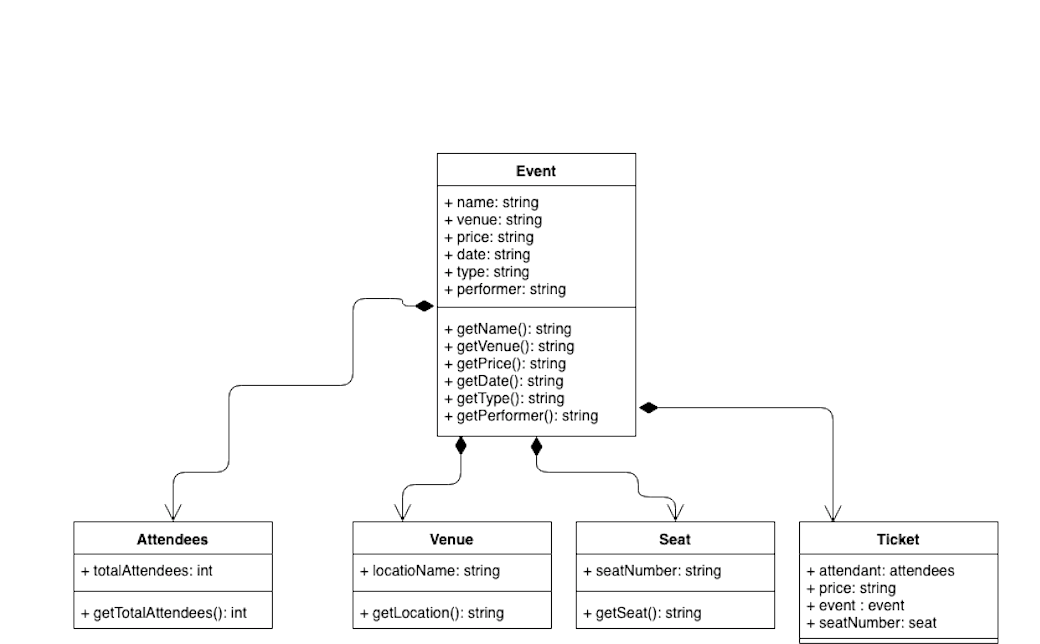
****

**Figure 3.3 User Interface for Step Two of Ticket Purchasing Component**

### Component ticket purchasing processing detail

A detailed algorithmic description for the ticket purchasing component is presented.

#### Design Class hierarchy for ticket purchasing



**Figure 3.3.a Class Diagram structure for the ticket purchasing component**

#### Restrictions/limitations for ticket purchasing

#### 3.3.3.2.1 Melodious payment system will only accept credit cards from the following companies: Visa, American Express, Discover, and Mastercard.

#### 3.3.3.2.2 Melodious payment system will accept debit and credit cards only, there will be no option for Apple Pay or cash payments.

#### Performance issues for ticket purchasing

Not applicable for this component.

#### Design constraints for ticket purchasing

3.3.3.4.1 This version is available in English only.

3.3.3.4.2 Ticket system must be able to pass its data to the Event class/system.

3.3.3.4.3 Ticket system must keep track of the number of seats available.

3.3.3.4.4 Ticket system must keep have a price associated with each event.

#### Processing detail for each operation of ticket purchasing

##### 3.3.3.5.1 The user will click a drop-down list of numbers and select a number of tickets, up to five (5) they wish to purchase and then will click the 'Purchase Tickets' button.

##### 3.3.3.5.2 After clicking the 'Purchase Tickets' button the user will then be redirected to the Payment page.

##### 3.3.3.5.3 The user enters their 16-digit credit card number, expiration date, and security code.

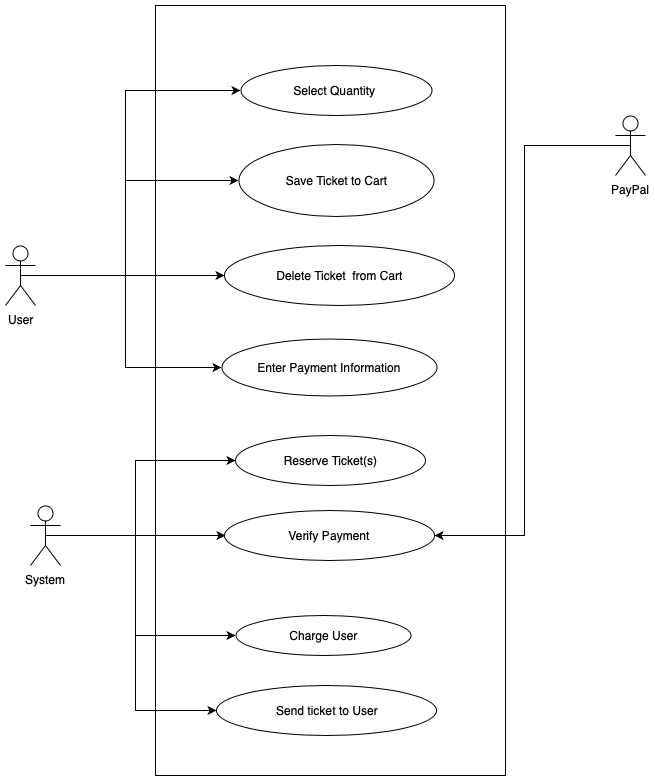
##### 3.3.3.5.4 After clicking the 'Next Step' button the user will then be redirected to the Shipping and Payment Confirmation page.

##### 3.3.3.5.5 The user will select their preferred shipping method and then click on the ‘Pay Securely Now’ to confirm their ticket purchase.

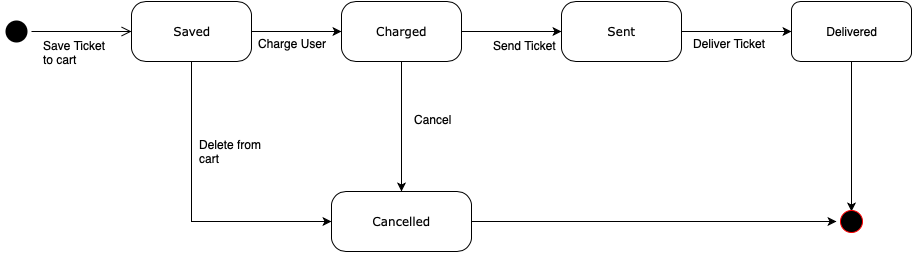
### Dynamic Behavior for Component Ticket Purchasing

Behavior for the ticket component interacts with the event class as a composition of the event class. Ticketing is one of the four compositions of the event class, therefore is indirectly interacting with the three other composition classes, venue, attendees, and seats.

### Interaction Diagrams



**Figure 3.3.b Use case diagram for Melodious Ticket Purchasing**

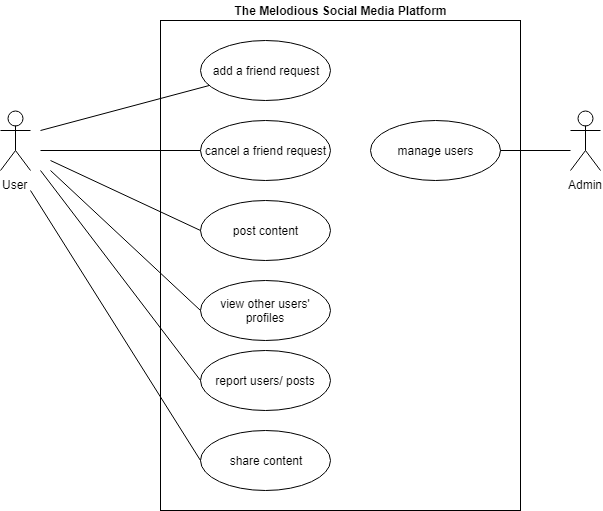


**Figure 3.3.c State diagram for Melodious Ticket Purchasing**

## Description for Component: Melodious Social Media Platform

Author - Linh Huynh

The Melodious Social Media platform is one of the most important component for the entire system because this is the main tool for users to connect with each other and share music interests. Figure 3.4.a is a use case diagram describing different features of the Melodious Social Media platform.



**Figure 3.4.a Use case diagram for Melodious Social Media Platform**

### Processing narrative for component Melodious Social Media platform

After logging in to the system, the user can navigate to the Melodious Social Media Platform. The user takes different actions to interact with other users. Features included in this platform are sending or cancelling friend requests, posting and sharing content, view others’ profiles, and report inappropriate contents. Depends on the actions that the user chooses, the system responses accordingly in real time and directs the user to the appropriate interfaces.

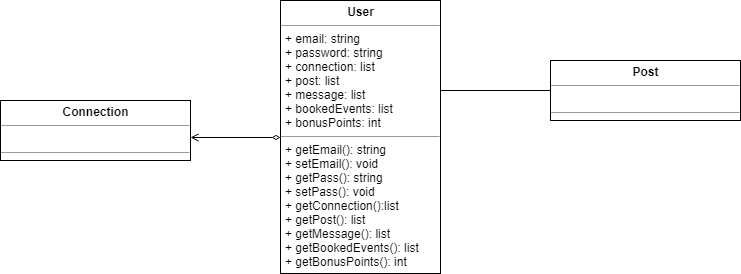
### Component Melodious Social Media Platform interface description

The Melodious Social Media system takes requests from users as inputs, sends them to the servers and produces data or actions as outputs.

### Component Melodious Social Media Platform processing detail

The component receives input from users and performs actions depending on the inputs. A detailed algorithmic description for each component is presented. Figure 3.4.b is a class diagram for the Melodious social media system.

#### Design Class hierarchy for component Melodious Social Media Platform



**Figure 3.4.b Class Diagram for Melodious Social Media Platform**

#### Restrictions/limitations for component Melodious Social Media Platform

3.4.3.2.1 Melodious Social Media platform can only be accessed by registered accounts.

3.4.3.2.2 The interactions between the users and the system are in real time. Specifically, the system must response to the users’ actions within 3 seconds.

#### Performance issues for component Melodious Social Media Platform

The user can only take one action at a time. For example, the user can only post a video after finishing sharing a song. Posting and sharing concurrently are impossible.

#### Design constraints for component Melodious Social Media Platform

3.4.3.4.1 The system is available in English only.

3.4.3.4.2 A post can only be 250 words in length.

#### Processing detail for each operation of component Melodious Social Media Platform

Several operations included in this component are managing friend lists, managing contents, sharing contents, reacting to posts and sending directing messages to other users.

##### 3.4.3.5.1 Processing narrative for each operation

Managing friend lists allow the users to connect with other users and follow others’ activities. The user A can view a profile of user B and send B a friend request. B receives a notification of the request and has a choice to accept or decline the friend request. If B accepts it, the system will connect A and B. Each of them will see the other’s name in their friend lists and they can send direct messages to each other. If B declines the request, A will receive a notification that the request is declined.

Managing contents allow users to create, update, and delete their own posts such as songs, videos, or statuses. Depending on which actions the users take, the system will respond accordingly.

The user can also share posts from other users by hitting the share button of each post. The system then allows the user to simultaneously share on Melodious, Facebook, or Twitter. The user can choose one or multiple options and the post will be shared to their profiles on those platforms.

The system also allows users to react on each other’s posts. Users can like or comment posts.

The user can send direct messages to another user. The system forwards the message from user A to user B who will receive it immediately and respond to the user A. The conversation and the system’s execution will continue unless A and B stopped communicating.

##### 3.4.3.5.2 Algorithmic model for each operation

Managing friend lists:

Add

Receive notifications

Accept/ Decline

Managing contents

Create/ Update/ Delete

Sharing contents

Hit a share button

Choose platform(s) to share

Posts will be shared

Sending direct messages

Send messages

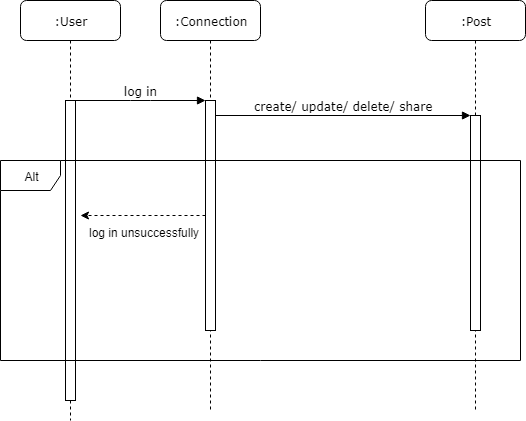
Delete messages

### Dynamic Behavior for Component Melodious Social Media Platform

After connected with the system, users can access to different features and create posts to interact with other users. The input from the users is processed and transformed into desired data or actions.

### Interaction Diagram

Figure 3.4.c is a sequence diagram of the Melodious social media platform that describe the interactions happening in the system.



**Figure 3.4.c Sequence diagram of Melodious Social Media Platform**

# User interface design

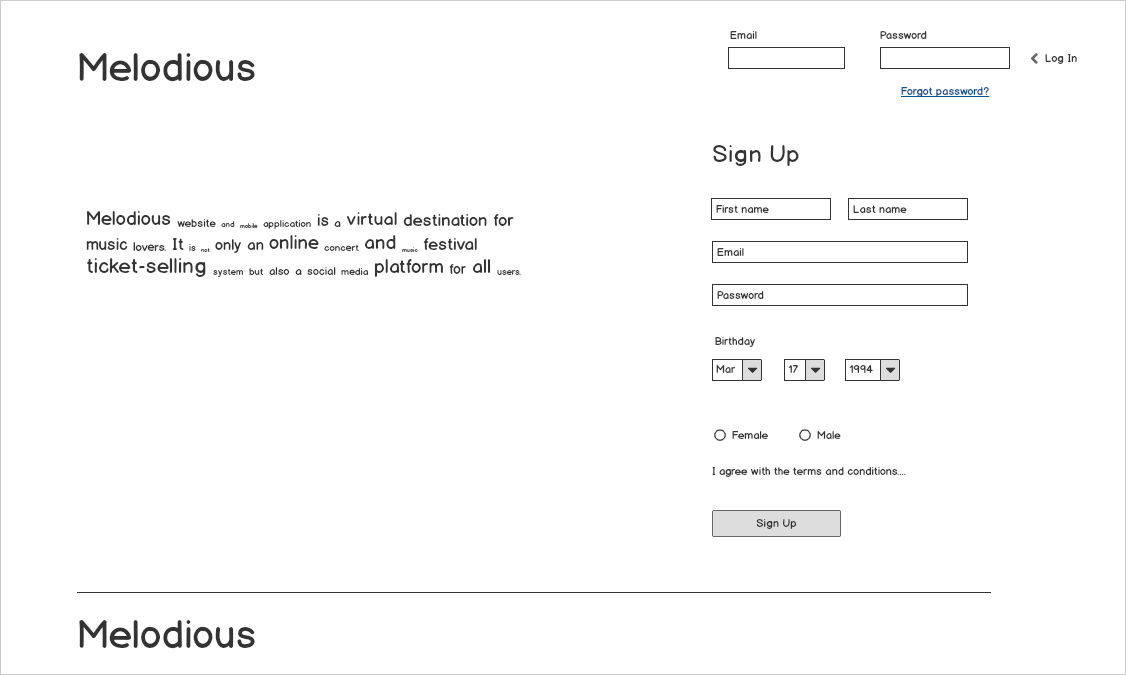
This section presents a description of the user interface design of the Melodious system, describes the user interface, demonstrates the relationships of different screens (below), and presents the purposes of each screen.

## Description of the user interface

A detailed description of the user interface including screen images or prototype is presented.

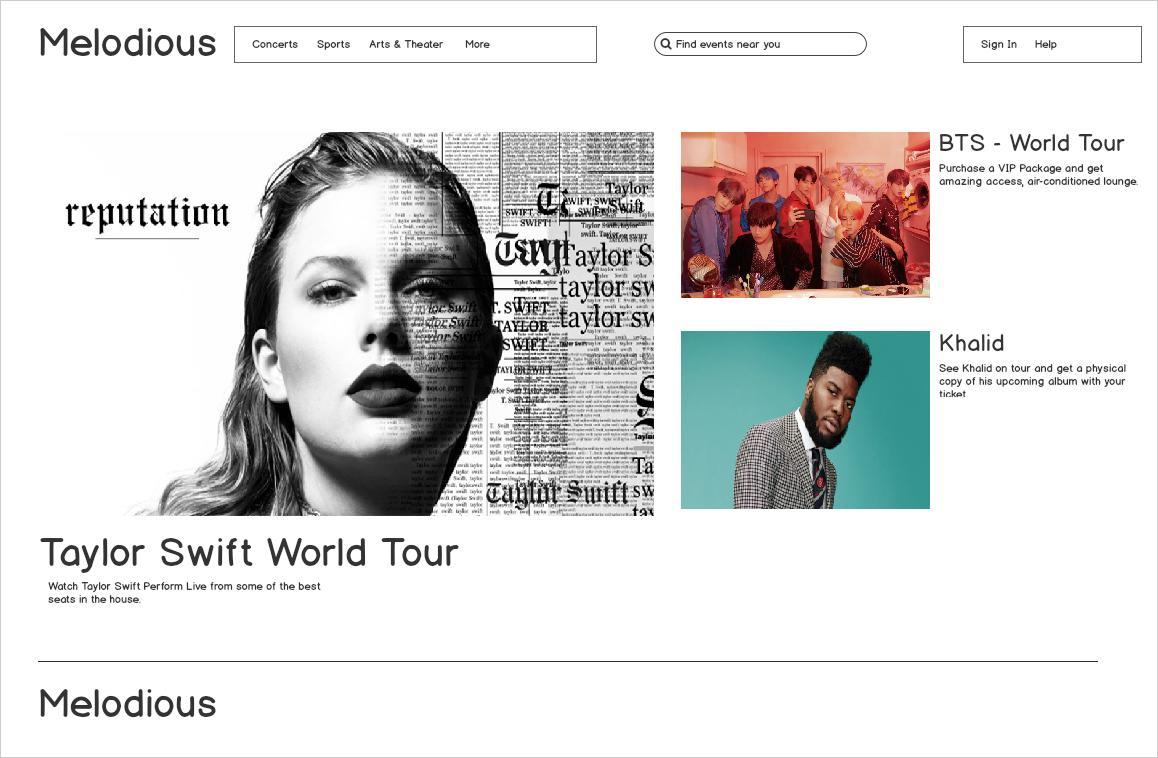
### Screen images

Representation of the interface from the user's point of view.



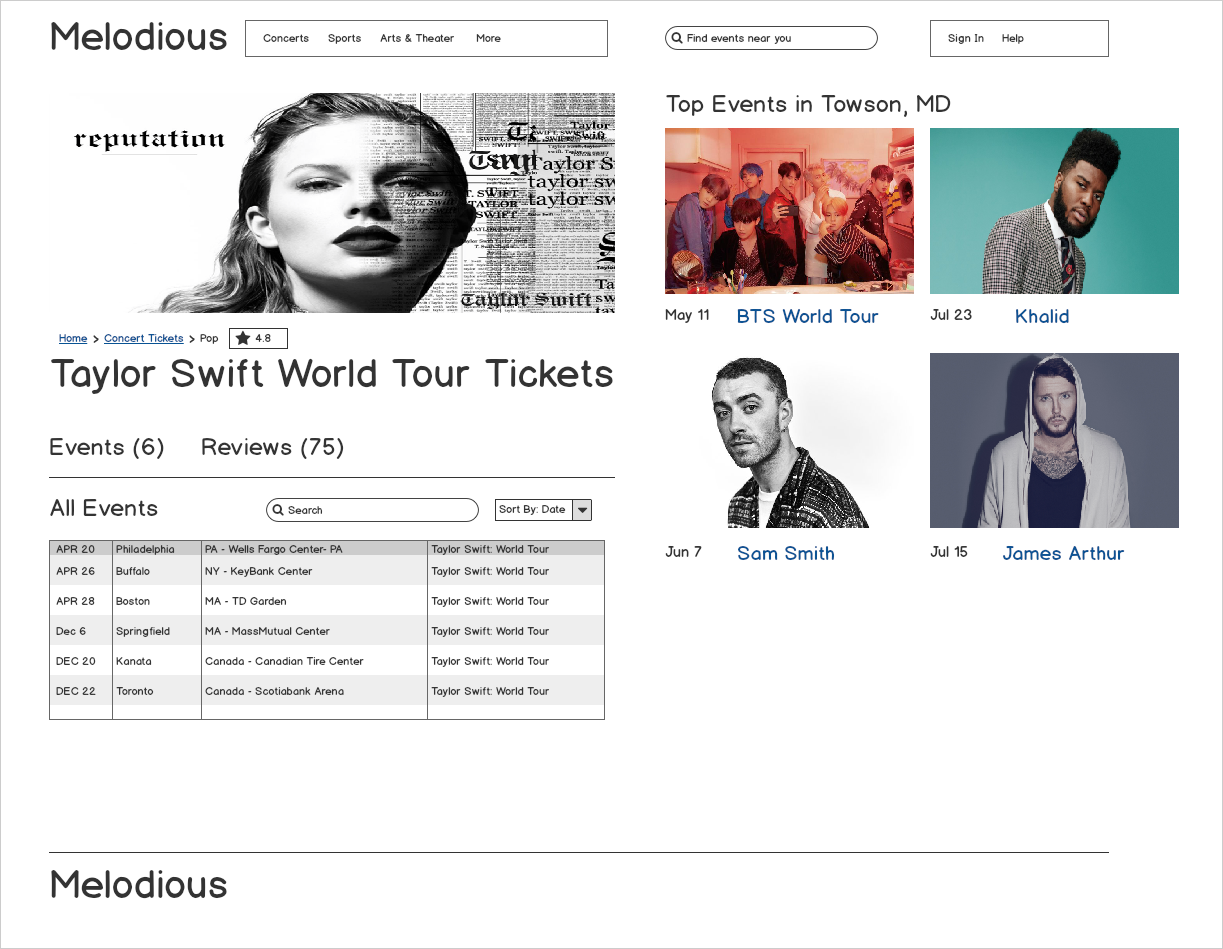
**Figure 4.1.1.a Melodious Login/ Registration Page**

**Description and Behavior:** Users wanting to sign in or register to the Melodious website are directed to the Login/Registration page. In order to register to the Melodious system, the user will fill out the sign-up form with their first name, last name, email, password, sex, and date of birth, check terms and conditions and submit the form by clicking the sign-up button. User with the Melodious account can sign in to the system by filling out their email and password and click the login button.

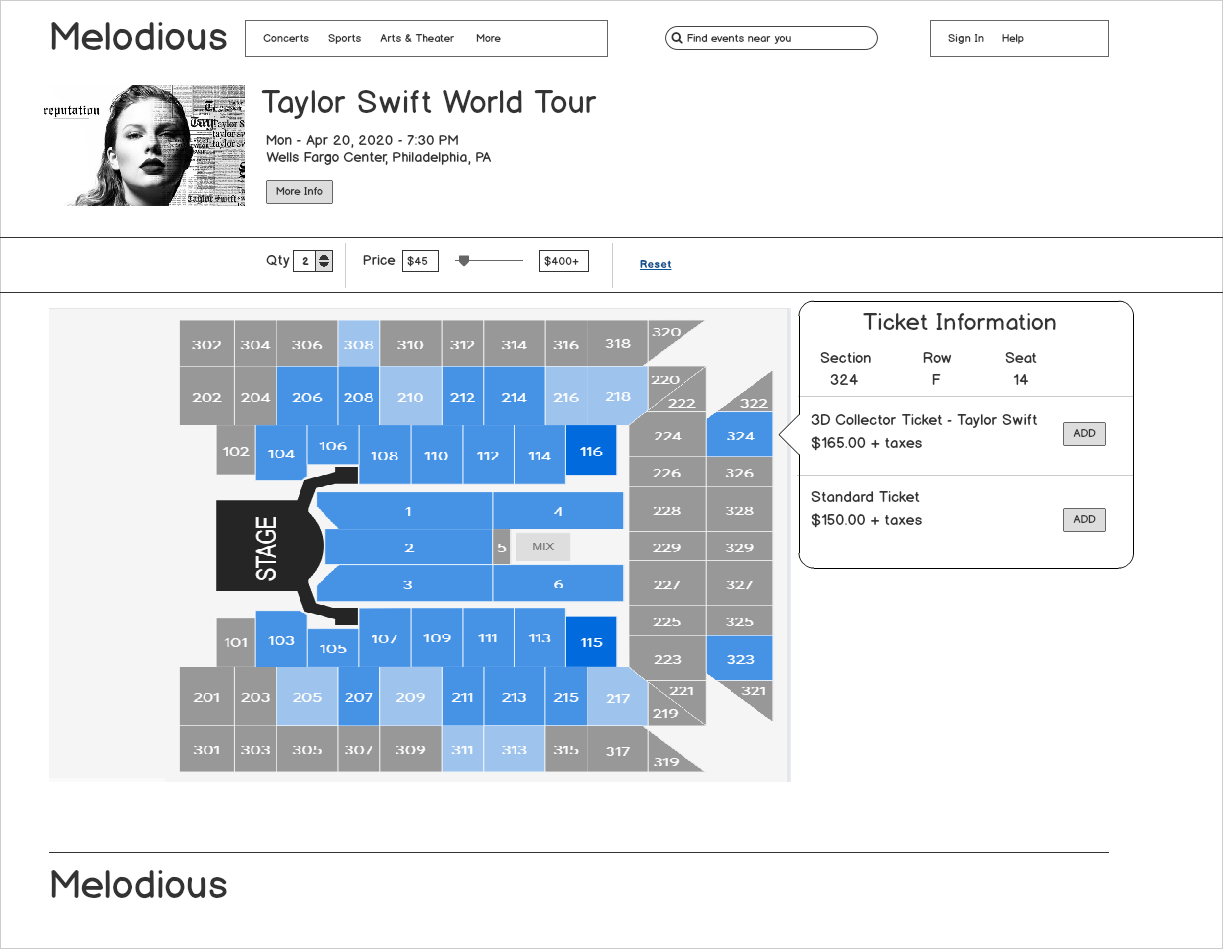


**Figure 4.1.1.b Welcome Page**

**Description and Behavior:** This is the landing page or welcome page for the Melodious website. A user does not have to be logged in to the Melodious system to access this page. The page consists of upcoming events, popular celebrities tours, or hot trending events. The tabs provide different types of events for focused users. User can use the search bar to search for the events near the user. User can use the sign in button to sign in to their account. User can click the help button to get to the help page where user can find FAQ, submit tickets, and find contact information.

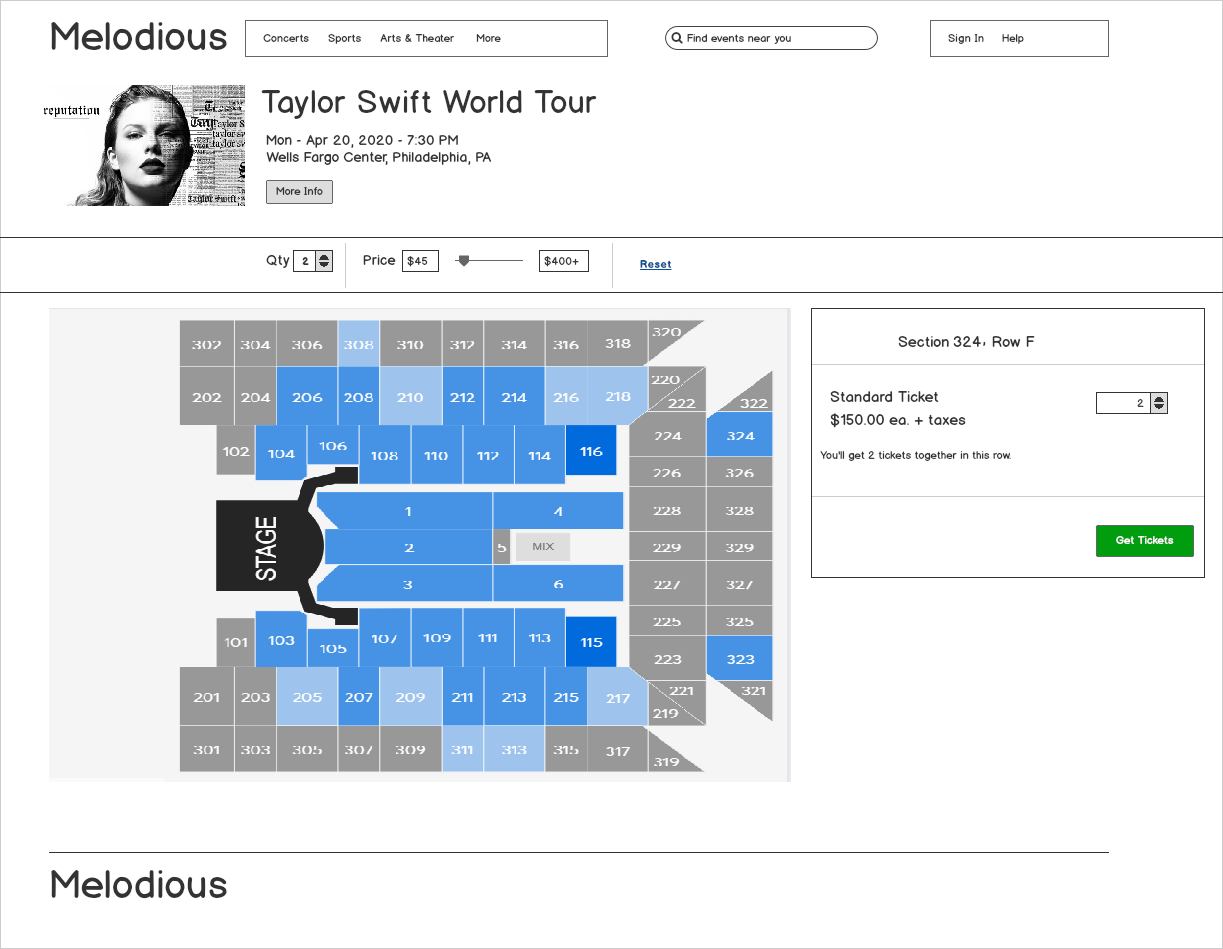
**Figure 4.1.1.c Events List Page**

**Description and Behavior:** This is the events list page for the Melodious website. A user does not have to be logged in to the system to access this page. The list of events is displayed in a table with the dates, location, venue, and name of the event. User can check the reviews posted by other users. The table row is clickable and the user can click on any event in the table to view more information.

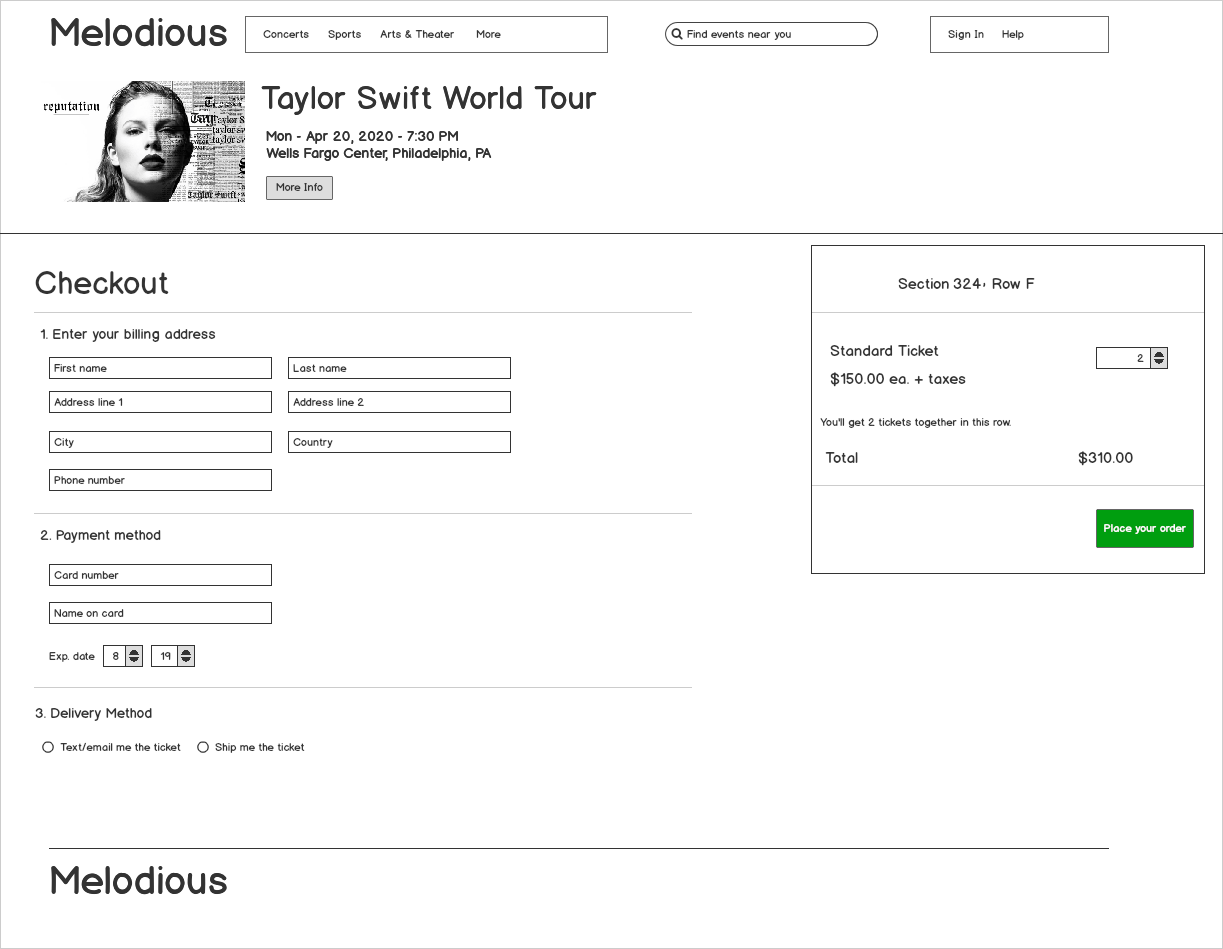
****

**Figure 4.1.1.d Ticket Info Page**

**Description and Behavior:** When a user clicks on an event to get ticket or more information, user is navigated to the tickets info page of the Melodious website. Tickets info page displays stage orientation, the blue highlighted rows show available seats, and the grey highlighted rows indicate seats not available. When a user clicks on the selectable section, ticket information pop-ups and the ticket information contains section number, row, seat, and tickets type. User can click the ‘Add’ button to add the ticket to purchase it.

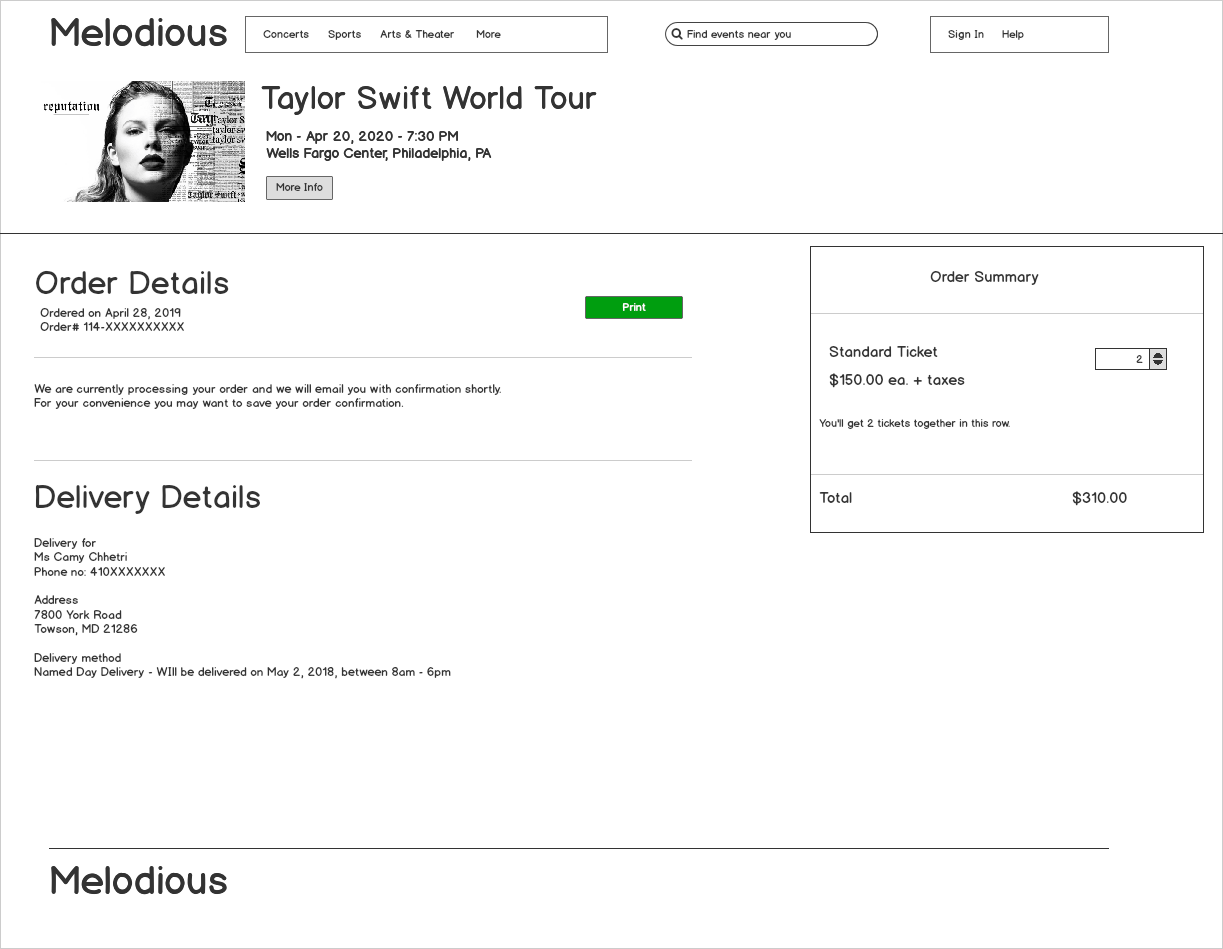
**Figure 4.1.1.e Ticket Purchase Page**

**Description and Behavior:** When the user clicks ‘Add Ticket’ button, user is directed to the ticket purchase page. The box provides ticket information and user can change the number of tickets. When the user is ready, user can click on the ‘Get Tickets’ button.

****

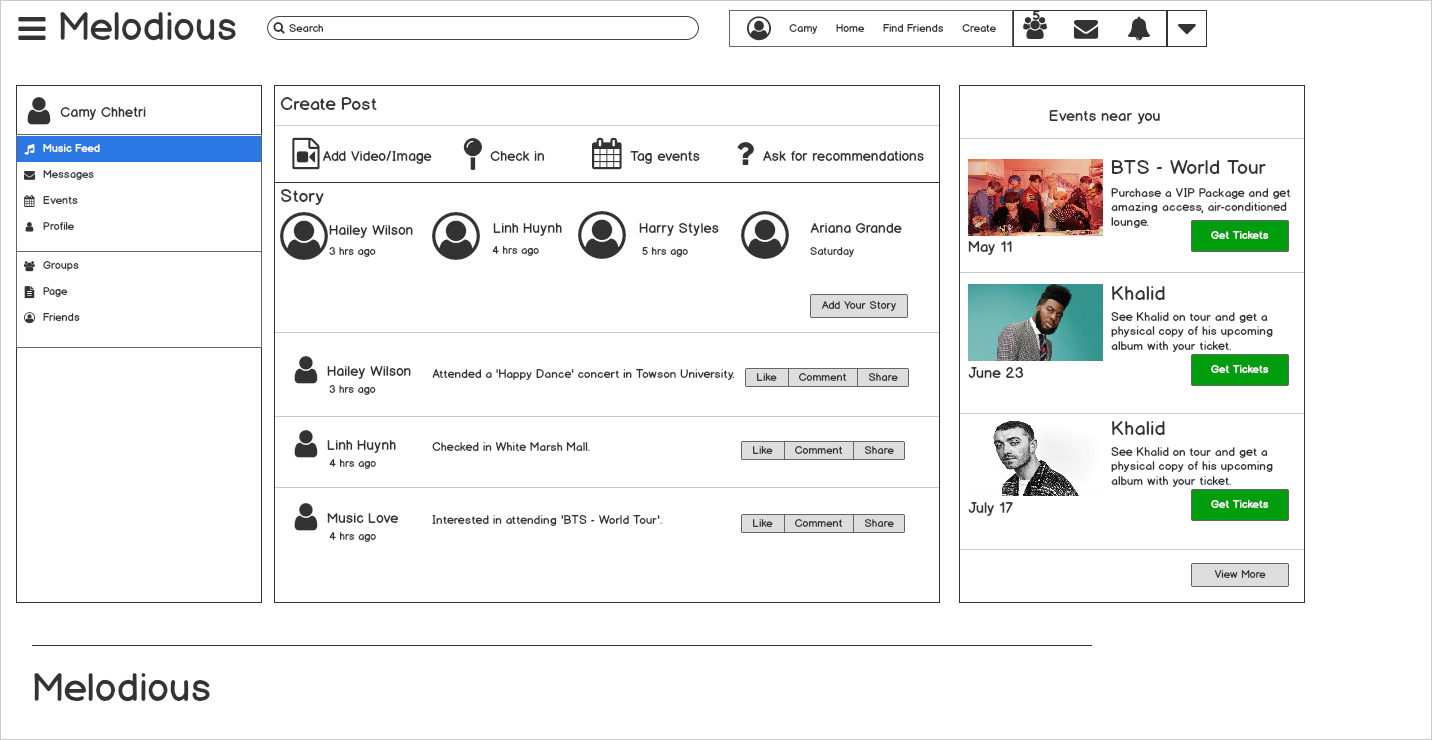
**Figure 4.1.1.f Ticket Payment Page**

**Description and Behavior:** When the user clicks ‘Get Tickets’ button, user is sent to the ticket payment page. In this page, user fills out their billing address, payment information, and delivery method. To submit the information and confirm the purchase, user clicks the ‘Place Your Order’ button to confirm their purchase.

****

**Figure 4.1.1.g Ticket Order Details Page**

**Description and Behavior:** When the user clicks ‘Place Your Order’ button, user is sent to the ticket order details page. This page provides the order details such as ordered date and delivery date, the order summary with the total, delivery details with the estimated delivery date. User may print their order details by clicking ‘Print’ button.

****

**Figure 4.1.1.h User Dashboard**

**Description and Behavior:** When a user successfully signs in to the Melodious system, the user is sent to their dashboard. The user dashboard page has news feed, messaging, events, profile, groups, page, and friends. News feed is the default action for the dashboard page. In the news feed page, user can create post, share story, view other people’s stories, like, comment, and share, and buy tickets. User can message their connection, add people to be their connection by sending request, accept/decline connection request, view their profile, and modify their profile and account settings.

### Objects and actions

All screen objects and actions are identified.

## Interface design rules

4.2.1 User Interface shall use the informational components such as tooltips, icons, progress bar, toast, confirmation dialogs, messages boxes, and modal windows.

4.2.2 The user interface shall be compatible with any browser such as Google Chrome, Mozilla, Internet Explorer, or Netscape Navigator by which user can access the system.

4.2.3 The user interface shall abide by W3C standards.

4.2.4 The menu shall be used when there are more than 5 options.

4.2.5 The dialogs shall be used to inform users about critical information which may involve users’ actions.

4.2.6 The banners shall be used to inform users about medium priority information.

4.2.7 The snack bar shall be used to inform users about low priority information such as confirmation, user feedback, or task status.

4.2.8 The tabs shall be used to allow navigation between groups of content.

4.2.9 The pagination shall be used to provide back and forth navigation.

4.2.10 The user interface shall contain appropriate content labeling for those who experience a text-only version.

4.2.11 The user interface shall be flexible, responsive layouts which content scale in relation to the screen size.

4.2.12 The text style shall follow material design typography guidelines.

**4.2.13 Colors**

4.2.13.1 The user interface shall use #6200EE as the primary color.

4.2.13.2 The user interface shall use #03DAC5 and #3700B3 as the secondary colors.

4.2.13.3 The user interface shall use #FFFFFF for the background.

4.2.13.4 The user interface shall use #B00020 as the error color.

## Components available

4.3.1 User Interface shall use the input controls such as buttons, input fields, checkboxes, menus, radio buttons, dropdown lists, list boxes, toggles.

4.3.2 User Interface shall use the navigational components such as breadcrumb, slider, search field, pagination, slider, tags, icons.

## UIDS description

The user interface development system is not available.

# Restrictions, limitations, and constraints

Special design issues which impact the design or implementation of the software are noted here.

5.1 The system shall incorporate CapLinked, a digital rights management software, to protect copyright of videos uploaded by uprising artists.

5.2 To register as an artist, a user must complete an artist registration form and provide required information to verify identity.

5.3 A post can only be 250 words in length.

5.4 The system is available in English only.

5.5 The system shall utilize Microsoft Azure SQL database to store all data.

**5.6 Security constraints**

5.6.1 The system shall encrypt all credit card information.

5.6.2 The system shall encrypt all password information.

5.6.3 User will be automatically redirected back to homepage if they stay at the payment screen for more than 20 minutes.

**5.7 Web-based system**

5.7.1 The system shall operate on Chrome version 48.0.2564 and up.

5.7.2 The system shall operate on Firefox version 44 and up.

5.7.3 The system shall operate on Microsoft Edge version 20.10240 and up.

5.7.4 The system shall operate on Safari 9 and up.

5.7.5 The system shall operate on Samsung Internet version 4 and up.

**5.8 Mobile application**

5.8.1 The system shall operate on Android version 5.0 and up.

5.8.2 The system shall operate on iOS version 10.1 and up.

5.8.3The system shall be hosted on AWS Amplify.

# Testing Issues

The Melodious system will be tested using behavior testing techniques to validate if the software meets the specified requirements and customer’s needs.

## Classes of tests

The types of tests to be conducted are specified, including as much detail as is possible at this stage. Emphasis here is on black-box and white-box testing. The table 6.1 below describe the test classes, expected responses, performance bounds, and identification of critical components of the Melodious system.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Class** | **Description** | **Expected Response** | **Performance Bounds** | **Components** | **Critical** |
| User Registration | Validation of successful user registration to the system | The system registers the user to the system and gives user privileges. | User should meet the age requirement, provide a valid email address. | User | Yes |
| Artist Registration | Validation of successful artist registration to the system | The system registers the user as the artist and gives artist privileges. | The artist must own at least 10 songs or music videos. | Artist | Yes |
| User Login | Validation of successful user log-in to the system | The system logs the user in and shows pages accessible by the user. | Unique username and password are required. | User | Yes |
| Artist Login | Validation of successful artist log-in to the system | The system logs the artist in with artist privileges. | Unique username and password are required. | Artist | Yes |
| Accept Friends Request | Validation of successful user added to the friend list. | The system adds the accepted user to the friend's list. | The friend request must be sent first in order to accept it. | Users |  |
| Add Friends | Validation of a successful friend request invites to send to the user. | The system sends the friend request invite to the user. |  | Users |  |
| Decline Friend Request | Validation of a successful friend request removal. | The system removes the friend request. |  | Users |  |
| Purchase Ticket | Validation of successful ticket purchase. | The system processes the order. | The payment information must be valid. | Users | Yes |

**Table 6.1 Test classes of the Melodious system**