

Software Requirements Specification

Melodious System

Version 1.5

Prepared by musicLovers

(Linh Huynh, Camy Chhetri, and Hailey Wilson)

AIT624 - COSC612

Professor: Dr. Josh Dehlinger

04/19/2019

**Table of Content**

[**1. Introduction**](#_1t3h5sf) **3**

[1.1 Purpose](#_4d34og8) 3

[1.2 Document Conventions](#_2s8eyo1) 3

[1.3 Intended Audience and Reading Suggestions](#_17dp8vu) 4

[1.4 Product Scope](#_3rdcrjn) 4

[1.5 References](#_26in1rg) 4

[**2. Overall Description**](#_lnxbz9) **5**

[2.1 Product Perspective](#_35nkun2) 5

[2.2 Product Functions](#_1ksv4uv) 6

[2.3 User Classes and Characteristics](#_44sinio) 8

[2.4 Operating Environment](#_2jxsxqh) 8

[2.5 Design and Implementation Constraints](#_z337ya) 9

[2.6 User Documentation](#_3j2qqm3) 9

[2.7 Assumptions and Dependencies](#_1y810tw) 9

[**3. External Interface Requirements**](#_4i7ojhp) **9**

[3.1 User Interfaces](#_2xcytpi) 10

[3.2 Hardware Interfaces](#_1ci93xb) 12

[3.3 Software Interfaces](#_3whwml4) 12

[3.4 Communications Interfaces](#_2bn6wsx) 13

[**4. System Features**](#_qsh70q) **13**

[4.1 Artist Registration](#_3as4poj) 14

[4.2 Log-in System](#_1pxezwc) 17

[4.3 User Registration Features](#_49x2ik5) 21

[4.4 Prioritization Algorithm](#_2p2csry) 24

[**5. Other Nonfunctional Requirements**](#_147n2zr) **24**

[5.1 Performance Requirements](#_3o7alnk) 24

[5.2 Safety Requirements](#_23ckvvd) 25

[5.3 Security Requirements](#_ihv636) 25

[5.4 Software Quality Attributes](#_32hioqz) 25

[5.5 Business Rules](#_1hmsyys) 25

[**6. Other Requirements**](#_41mghml) **26**

[6.1 Third-party system requirements](#_2grqrue) 26

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Linh Huynh  Camy Chhetri  Hailey Wilson | 4/10/2019 | Create a draft for the documentation  Add diagrams and use case descriptions | 1.1 |
| Hailey Wilson | 4/12/2019 | Update draft to accommodate the suggested changes after in class peer-review | 1.2 |
| Linh Huynh | 4/15/2019 | Add a table of content, format page layout, reorganize diagrams, add prioritization algorithm | 1.3 |
| Hailey Wilson | 4/16/2019 | Update draft to accommodate the suggested changes after professor review | 1.4 |
| Camy Chhetri | 4/18/2019 | Finalize SRS | 1.5 |

# 

# 

# 

# 

# 

# 

# 

# **1.** **Introduction**

## 1.1 Purpose

According to Billboard, approximately 32 million people attend at least one music festival in the U.S annually. They purchase concert tickets mainly through ticket sales portals such as Ticketmaster and StubHub. To share our music interests and follow their favorite artists, they utilize multiple social media platforms including Facebook, Instagram, SnapChat, and Twitter.

Wouldn't it be amazing if there is a website or mobile application dedicated solely to the music community for purchasing tickets, connecting with music lovers, sharing their music experience, and promoting new artists? Well, look no further because the Melodious website is a one-stop shop for the music community. The Melodious website and mobile application are a virtual destination for music lovers. It is not only an online concert and music festival ticket-selling system, but also a social media platform for all users. The Melodious application is unique for multiple reasons. One, it targets users who have a strong interest in music events and want to connect with others through music. Two, for every ticket a user purchases through Melodious they receive points which go towards discounts on future tickets they purchase. Three, it is also a social media platform where users can send friend requests, invite each other to concerts, send messages, share and upload music sheets as well as videos. Each user has a personal profile that includes music interests, instrument, and concerts attended. Lastly, users can watch concert videos from the past.

The system integrates information of all small and large-scale concerts that occur not only in the U.S but also around the world. Melodious provides a platform for uprising artists to plan and advertise their concerts. Uprising artists and composers can have a greater reach to audiences by posting concert details such as locations, date, time, and the number of seats. Then, the system sells those tickets to other users and earns commission from every ticket sold. The final product of this project is a system that integrates three main features: selling tickets with the points system, building connections among music lovers, and promoting new artists.

## 1.2 Document Conventions

Acronyms

SRS – Software Requirements Specification

UI - User Interface

PIN - Personal Identification Number

## 1.3 Intended Audience and Reading Suggestions

Important stakeholders of Melodious include users, artists, project sponsor, project manager, and software engineers. This Software Requirements Specification (SRS) document has been written up for our many different stakeholders who will be interacting with the Melodious system. Software developers will be able to read this document and understand what needs to be implemented to satisfy system requirements. The SRS document is important for customers who are not directly involved in the development of this system, such as project managers, as it will ensure the system being documented will meet their need in a satisfactory way. Developers and customers alike should read this document and be able to understand the system requirements. In doing so, this document will help ensure the needs of the system has been met satisfactorily.

The current SRS provides an introduction to the system, provides an overall description of the system, the external interface requirements, system features, and nonfunctional requirements for the Melodious system. Section 1 of the SRS gives a detailed introduction to the project which includes the purpose, the intended audience, and the product scope. Section 2 summarizes the overall description of the system, including the product perspective, product functions, operation environment, constraints on design and implementation, user documentation, and assumptions and dependencies. This section provides a class diagram to show the design of the objects, and their relationships to each other within this system. Section 3 gives the external interface requirements; these interfaces include the user, hardware, software, and communications interfaces. Section 4 will provide the main system features to be implemented in this system. Lastly, Section 5 provides a detailed list of the nonfunctional requirements describing the operational domain of this system including the performance requirements, security requirements, software quality attributes, and business rules.

## 1.4 Product Scope

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.

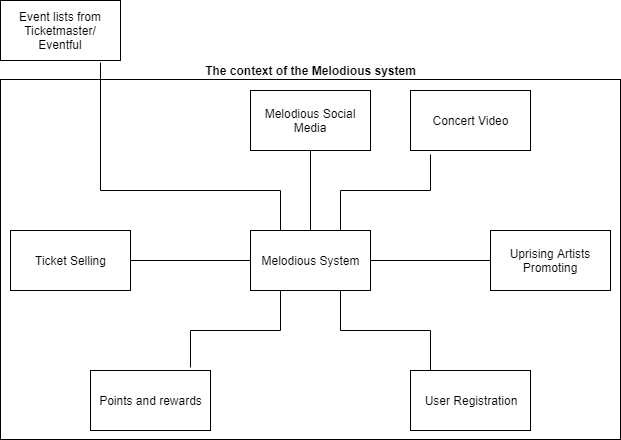
## 1.5 References

Smith, B. (2016, May 18). I/O 2016: Our Definitive Guide to Design. <https://medium.com/google-design/i-o-2016-our-definitive-guide-to-design-4d64d2db7150>

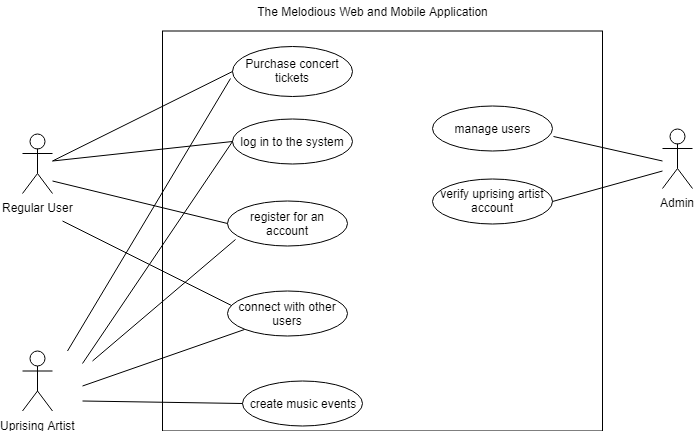
# **2.** **Overall Description**

## 2.1 Product Perspective

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. Many ticket selling platforms such as Ticketmaster or StubHub have significant issues that are not beneficial for consumers and the music industry. First, those platforms focus on not only medium- to large-scale concerts but also other events including sport and theaters, which does not promote amateur artists. Second, other platforms charge excessive fees including transaction fee, processing fee, convenience fee, and servicing fee. Third, they allow overpriced resale tickets, which may lead to ticket inflation. Thus, the Melodious website and mobile application will eliminate all these issues and maximize benefits for not only music fans but also artists. Figure 2.1.a and 2.1.b are a context diagram and an overall use case diagram to describe different components of the Melodious system.



**Figure 2.1.a Context diagram for the Melodious system**

****

**Figure 2.1.b Overall use case diagram for the entire Melodious system**

## 2.2 Product Functions

The system combines eight major groups of functions as described below. Figure 2.2.a describe the entities, attributes, and relationships of these components.

**2.2.1 Ticket buying component for all users**

2.2.1.1 The system shall allow all users to search for upcoming music events.

2.2.1.2 The system shall provide users with details of past and future music events.

2.2.1.3 The system shall allow all users to purchase verified concert tickets.

2.2.1.4 The system shall update the concert details near real-time if there are any changes in location, time, and date.

2.2.1.5 The system shall display the number of points, to be awarded, that is associated with each music event.

2.2.1.6 The system shall save and update points for user accounts after the transaction is complete.

2.2.1.7 The system shall include a user’s full name on the ticket information.

**2.2.2 Social media component for all users**

2.2.2.1 The system shall allow users to send friend requests.

2.2.2.2 The system shall allow users to receive friend requests.

2.2.2.3 The system shall allow users to invite others to music events.

2.2.2.4 The system shall allow users to share music interests and events with their connections.

2.2.2.5 The system shall allow users to manage their own dashboards and profile information.

**2.2.3 Regular users registration**

2.2.3.1 User shall be able to register using their name, birthdate, and email.

2.2.3.2 System shall send a verification email to registering user.

2.2.3.3 User shall receive a verification email to the provided email during registration.

**2.2.4 Uprising artists registration**

2.2.4.1 Uprising artists shall complete an online verification form.

2.2.4.2 The system shall check that the artist owns at least 10 songs or music videos uploaded by the artists in the registration page.

2.2.4.3 If the criteria in 2.2.4.2 matches, then the system shall create an account for uprising artists.

2.2.4.4 If the criteria in 2.2.4.2 do not match, then the system shall inform the user of the failure to sign up through email.

**2.2.5 All Users log-in component**

2.2.5.1 The system shall allow users to log in by providing a Username and Password at the Login Screen.

2.2.5.2 The system shall check the Username and Password provided to determine if the user exists in the system.

**2.2.6 Uprising artists event activities**

2.2.6.1 The system shall allow uprising artist accounts to create music events.

2.2.6.2 Uprising artists shall be able to add event details including name, date, price, music genre, and the number of available tickets.

2.2.6.3 The system shall allow uprising artists to post videos

2.2.6.4 The system shall allow uprising artists to post their performances.

**2.2.7 Admin activities**

2.2.8.1 Admin shall have access to profile information of all users.

2.2.8.2 Admin shall view the activities of all users.

2.2.8.3 Admin shall ban users temporarily if they violate the policies as stated in TBD.

2.2.8.4 Admin shall delete user accounts if they violate the policies as stated in TBD.

2.2.8.5 Admin shall send scheduled website maintenance message to all users.

2.2.8.6 Admin shall upload past concert videos to the system.

2.2.8.7 The system shall allow users to access a list of Q&A regarding technical support, event information, and refund process.

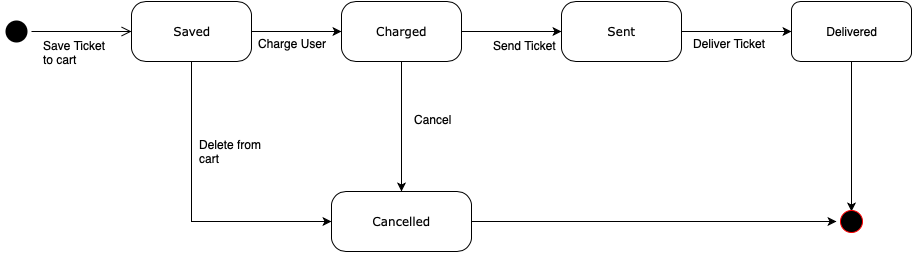
2.2.8.8 The system shall allow users to message Melodious staff if the users cannot find the answers in Q&A section.

**2.2.8 Concert video component**

2.2.8.1 The system shall allow users to watch past concert videos.

2.2.8.2 The system shall allow users to leave a comment in the videos.

2.2.8.3 The system shall allow users to rate the videos.



**Figure 2.2.b State diagram for Ticket selling system**

## 2.3 User Classes and Characteristics

According to the scope and goal of this project, there are three anticipated user classes that will use this product are regular users, uprising artists, and admin.

Regular users will be able to use all free services including the ticket buying component and social media component. Uprising artists can have access to all services of regular users and some additional features. They can promote themselves by posting their music products, creating music events, and selling tickets of their own events.

Admin is responsible for monitoring and controlling user accounts and user activities. Because the Melodious application includes a social media platform where users can freely connect with others and post their own content, it is important to keep track of user’s activities to prevent fraud. If a user reports about sensitive content or copyright violation, admin must review the report and decide to warn, ban, or delete an account within 24 hours. Admin is also in charge of answering any users’ questions regarding technical issues, event details, and refund process. They must understand all the procedures and policies of the system. They can view all users’ profiles, activities, and transactions. Based on the type of questions, the staff will check users’ past activities and provide solutions within 24 hours.

## 2.4 Operating Environment

**2.4.1. Web-based system**

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

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

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

2.4.1.4 The system shall operate on Safari 9 and up.

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

**2.4.2. Mobile application**

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

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

**2.4.3. Others**

2.4.3.1 The system shall be hosted on AWS Amplify.

## 2.5 Design and Implementation Constraints

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

2.5.2 The system is available in English only.

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

**2.5.4. Security constraints**

2.5.4.1 The system shall encrypt all credit card information.

2.5.4.2 The system shall encrypt all password information.

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

## 2.6 User Documentation

2.6.1 The documentation shall include a user manual for support personnel.

2.6.2 The documentation shall include a user manual for admin.

2.6.3 The system shall include an online tutorial that walk the users through all the features.

2.6.4 The system shall include a list of pre-defined Q&A for users.

## 2.7 Assumptions and Dependencies

2.7.1 It is assumed that all users using Melodious shall have one or more of the operating systems listed in 2.4.1 and 2.4.2.

2.7.2 It is assumed that new releases of third-party components incorporated in Melodious shall not affect on the system.

2.7.3 It is assumed that copyright law shall not change in the next 5 years.

2.7.4 It is a dependency that all third-party components incorporated in Melodious shall be Java compatible.

# **3.** **External Interface Requirements**

## 3.1 User Interfaces

**3.1.1 Components**

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

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

**3.1.2 Guidelines**

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

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

3.1.2.3 The user interface shall abide by W3C standards.

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

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

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

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

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

3.1.2.9 The pagination shall be used to provided back and forth navigation.

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

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

3.1.2.12 The text style shall follow material design typography guidelines.

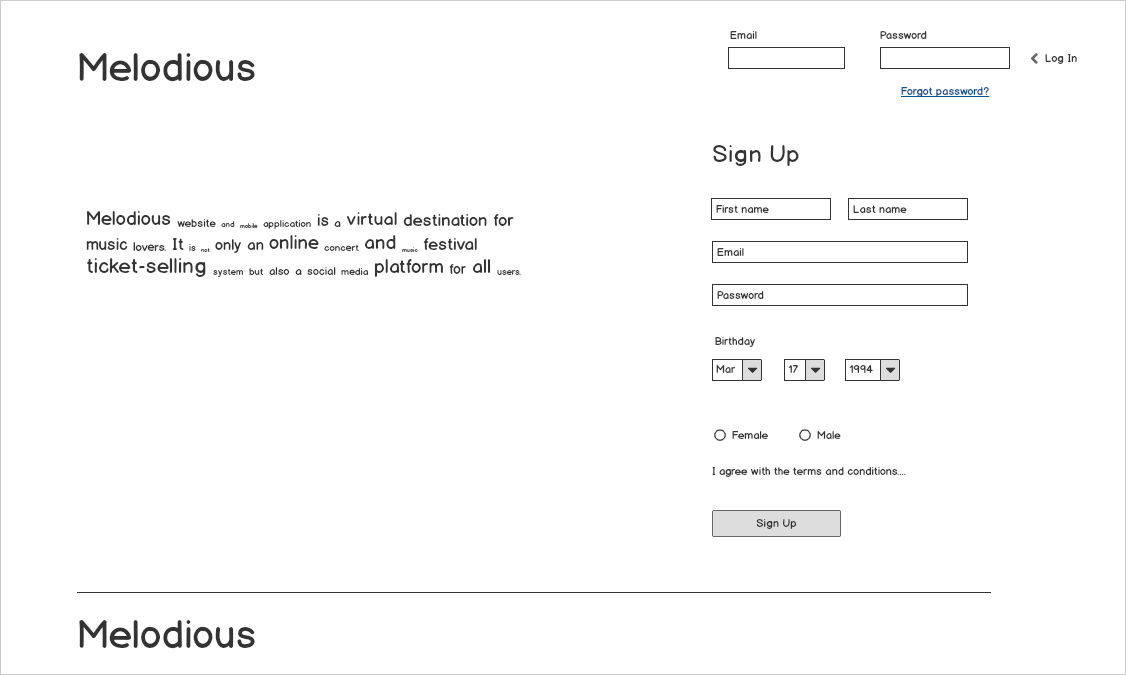
**3.1.3 Colors**

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

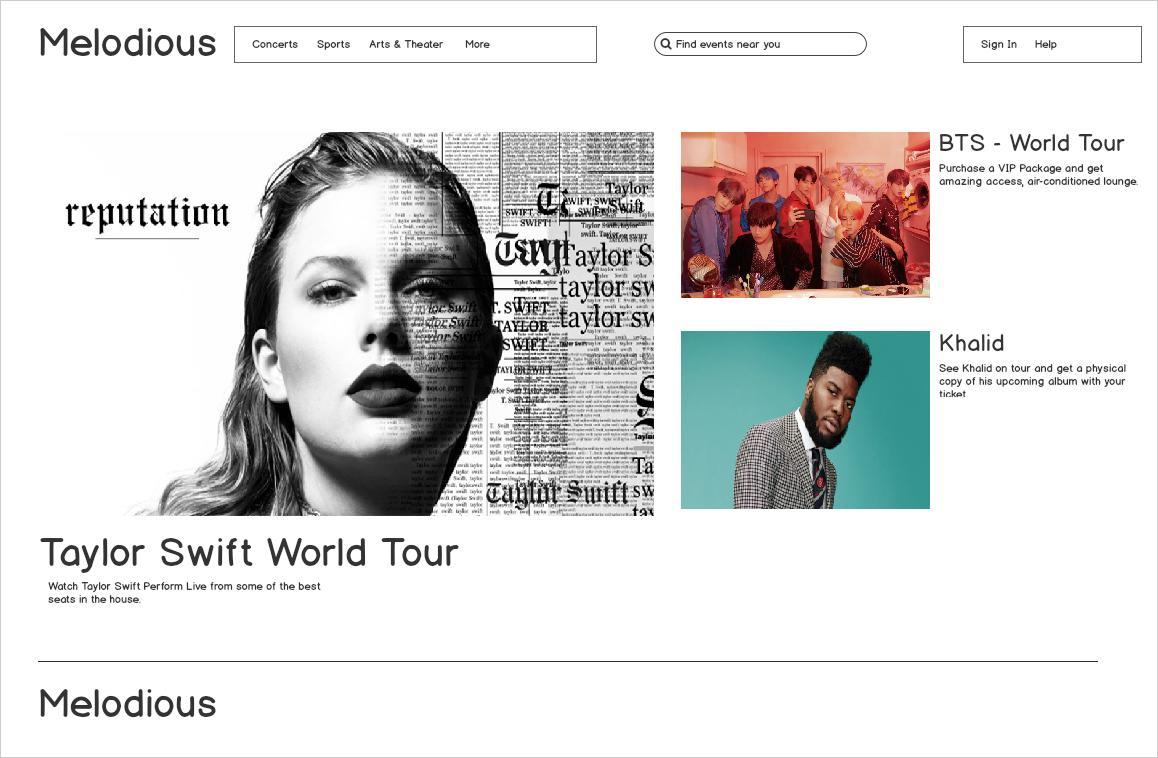
3.1.3.2 The user interface shall use #03DAC5 and #3700B3 as the secondary colors.

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

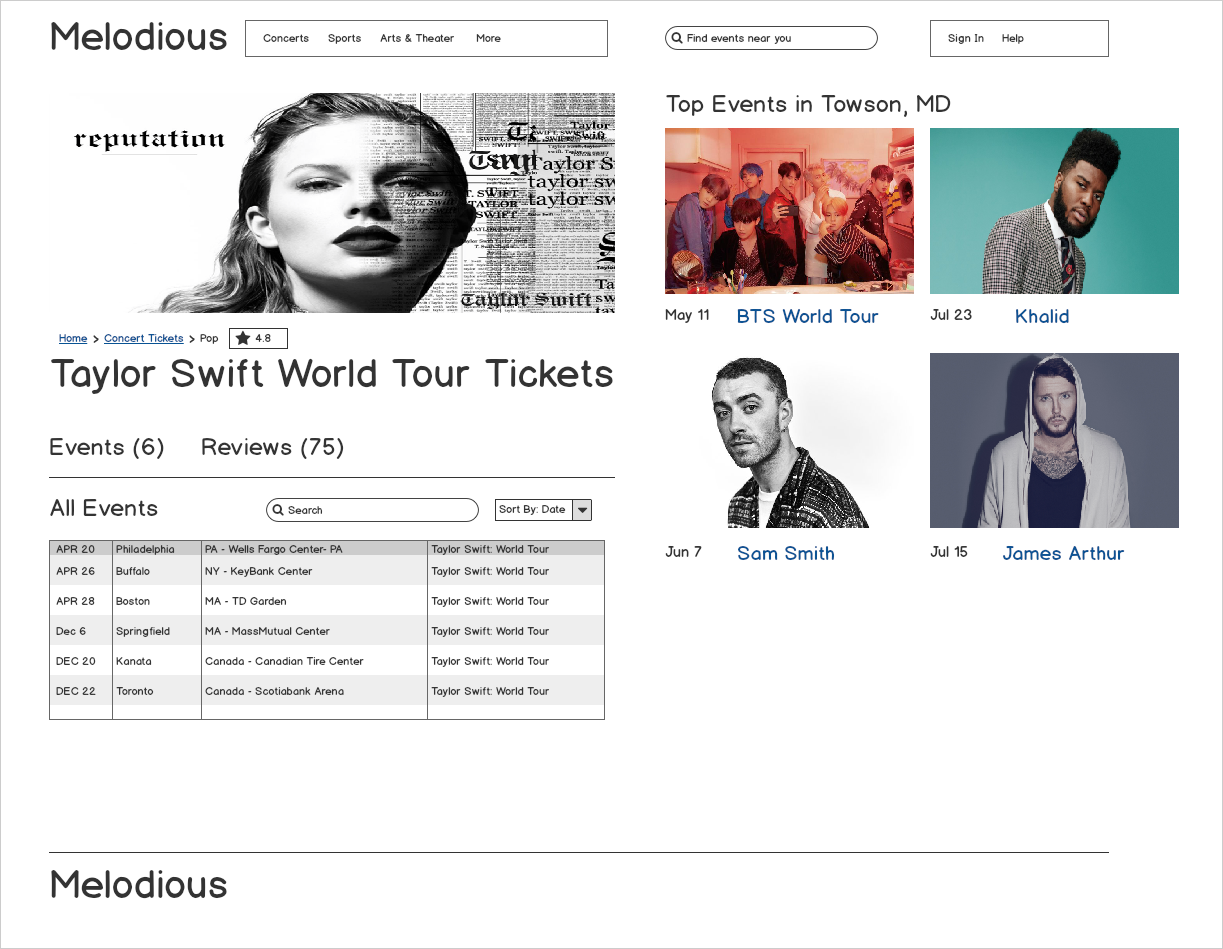
3.1.3.4 The user interface shall use #B00020 as the error color.



**Figure 3.1.a Melodious Login/ Registration Page**



**Figure 3.1.b Welcome Page**



**Figure 3.1.c Events Page**

## 3.2 Hardware Interfaces

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

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

## 3.3 Software Interfaces

3.3.1 The application shall run on C# 7.0 which is the latest version.

3.3.2 The backend program shall be implemented using C# programming language.

3.3.3 Visual Studio 2017 or Visual Studio Code shall be used to compile all the code that is developed in C#.

3.3.4 .Net Core 2.2, a cross-platform .NET shall be used to implement websites and apps on Windows, Linux, and macOS which is being used to create the application.

3.3.5 The database shall be deployed to Microsoft Azure SQL database.

3.3.6 The ticket booking system interface shall process payments, ticket bookings, orders.

3.3.7 The event handler system interface shall create events, update event, delete event, and schedule events.

3.3.8 The members manager shall manage users, their connections, messages, post, reviews, and comments.

3.3.9 The uprising stars manager shall communicate with event handler system to process the events created by the uprising stars.

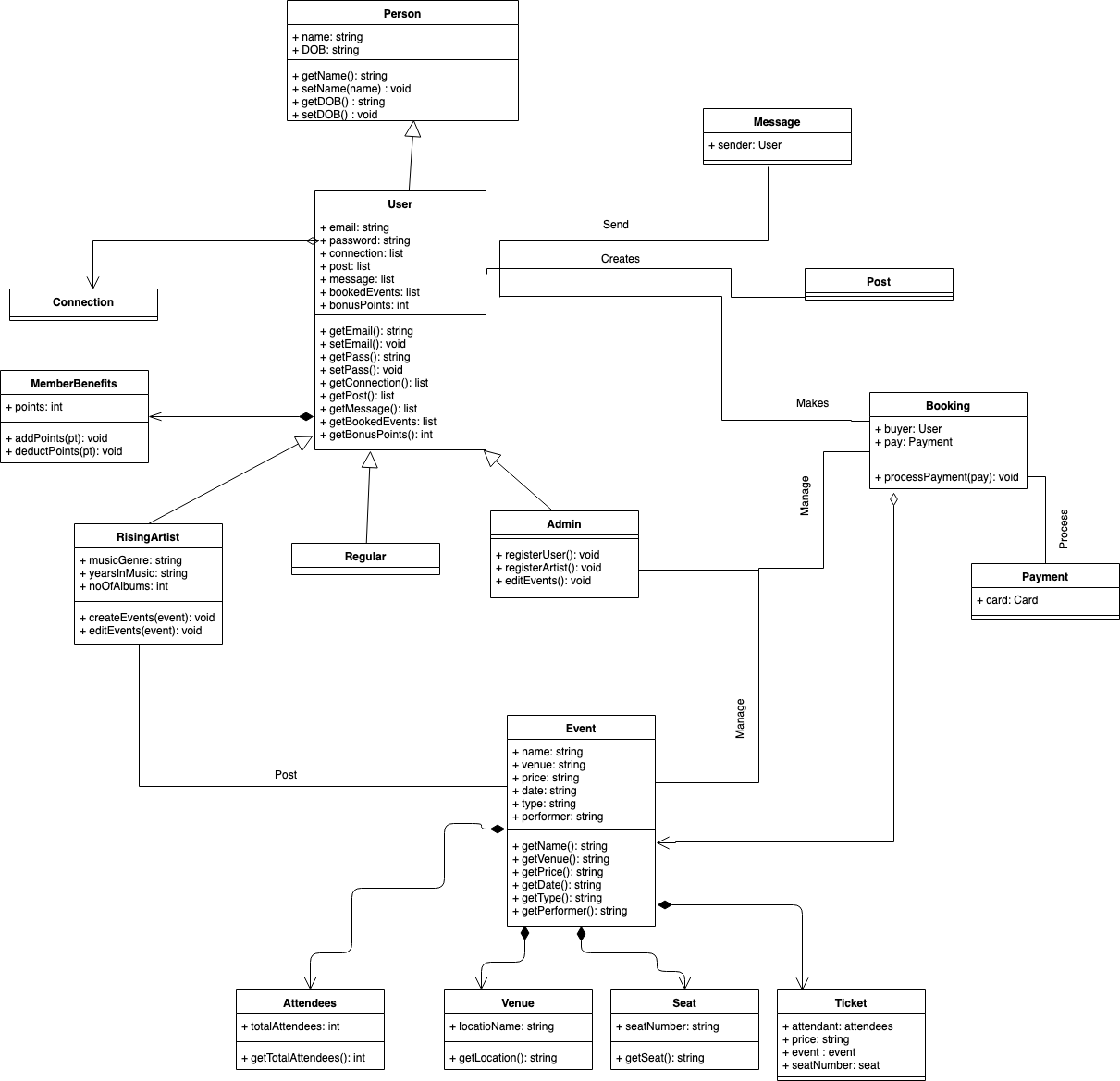
3.3.10 The uprising stars manager shall broadcast the event to all the users using the members manager.

## 3.4 Communications Interfaces

3.4.1 The melodious system shall use the HTTP protocol for communication over the internet and for the intranet communication will be through TCP/IP protocol suite.

3.4.2 The melodious systems shall connect to Eventful API and Ticketmaster API which provides upcoming events, concerts, and shows by keyword, location, time period, category, performer, or venue.

# **4.** **System Features**



**Figure 4.a Class diagram for the entire Melodious system**

## 4.1 Artist Registration

4.1.1 **Description and Priority**

The Melodious system allows artist to be able to register for a verified account. This will allow them to promote their events, connect with their fans and other artist as well. This system feature is of medium priority due to the fact that artist registration will not be the main system feature, but it does play an important role to the Melodious system as a whole.

4.1.2 **Stimulus/Response Sequences**

Artist will register for a verified account by filling out a verification form and an administrator or verification team will check the artist’s form.

4.1.3 **Functional Requirements**

REQ-1: The system shall provide an online verification form for the artist to complete upon their account creation.

REQ-2: The system shall provide an online verification form for the artist to complete upon their account creation.

REQ-3: The system shall send the administrator(s) an email with an attached verification form from the artist.

REQ-4: The system shall allow the administrator(s) to approve or deny the artist’s verification.

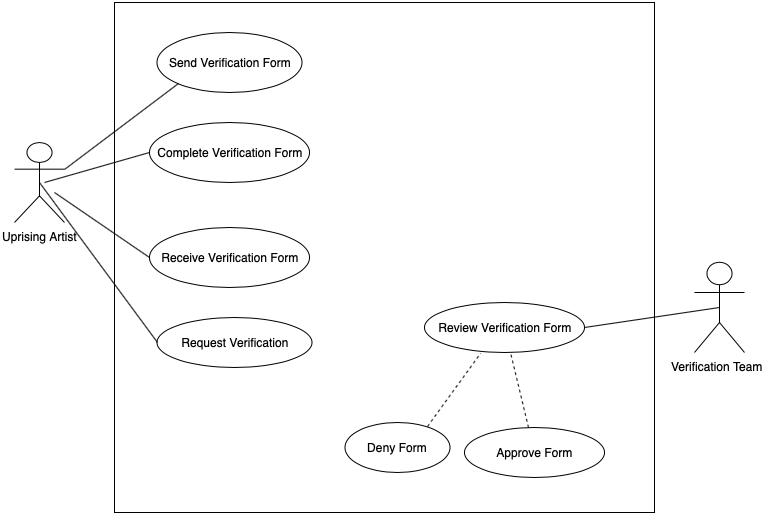
REQ-5: Upon approval of the verification form, the system shall grant the the artist a TBD symbol next to their name throughout the Melodious system

REQ-6: Upon denial of the verification form, the system shall send the artist an email informing them their verification form has been denied and stating the reasons why.

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 2.2.7 | | |
| Use Case Name: | Artist User Registration | | |
| Created By: | Hailey Wilson | Last Updated By: | Hailey Wilson |
| Date Created: | 04-10-2019 | Date Last Updated: | 04-10-2019 |

|  |  |
| --- | --- |
| Actors: | Artist, Verification Team, System |
| Description: | Artist is to register to melodious system using the verification form provided. |
| Trigger: | An artist wants to register as a verified artist on the Melodious system |
| Preconditions: | 1. The artist is on the registration page. |
| Postconditions: | 1. The administrator approves the verified artist registration request. |
| Normal Flow: | 1. Artist creates username and password. 2. Artist fills out the verification form. 3. Artist accepts the terms and conditions. 4. Artist submits the verification form. 5. System sends the artist verification form to the administrator(s). 6. Administrator(s) confirms the verification form. 7. System sends an email with the decision back to the artist. 8. System registers the artist as a verified artist in the system. |
| Alternative Flows: | 6. Administrator(s) denies the verification form.   * 6a1. System sends an email with the decision back to the artist. * 6a2. System shall not register the artist as a verified artist in the system, but as a regular user instead. |
| Exceptions: | The email format is not valid. |
| Includes: | 1. Create username and password 2. Fill out verification form 3. Accepts terms & conditions |
| Priority: | High |
| Frequency of Use: | Every time an artist wants to register as a verified artist to access the Melodious artist features. |
| Business Rules: | 1. The artist must owns at least 10 songs or music videos 2. The artist account will be able to create music events 3. The artist will be able to add event details including name, date, price, music genre, and the number of available tickets 4. The artists will be able to post videos or their performances 5. Password is encrypted |
| Special Requirements: | None |
| Assumptions: | None |
| Notes and Issues: | None |

**Table 4.1-1 Use case description of the artist registration**

****

**Figure 4.1.a Detailed use case diagram of the artist registration**

## 4.2 Log-in System

**4.2.1 Description and Priority**

The Melodious website and mobile application allows users to log in to the system using their usernames and passwords. This feature is a high priority because after login the users will have access to many free services and functions that the system provides as stated in section 2.2. Additionally, admin can monitor the number of users that are using the system and manage their activities.

**4.2.2 Stimulus/Response Sequences**

User want to have access to free services of the Melodious system

**4.2.3 Functional Requirements**

REQ-1: The system shall display a login page that asks for username and password

REQ-2: The users shall log in to the system using their username and password

REQ-3: The system shall validate the username and password

REQ-4: The system shall match the username and password with the database

REQ-5: The system shall connect user to the dashboard after successfully matching username and password with the database

REQ-6: The system shall display an error message if the user input a wrong username

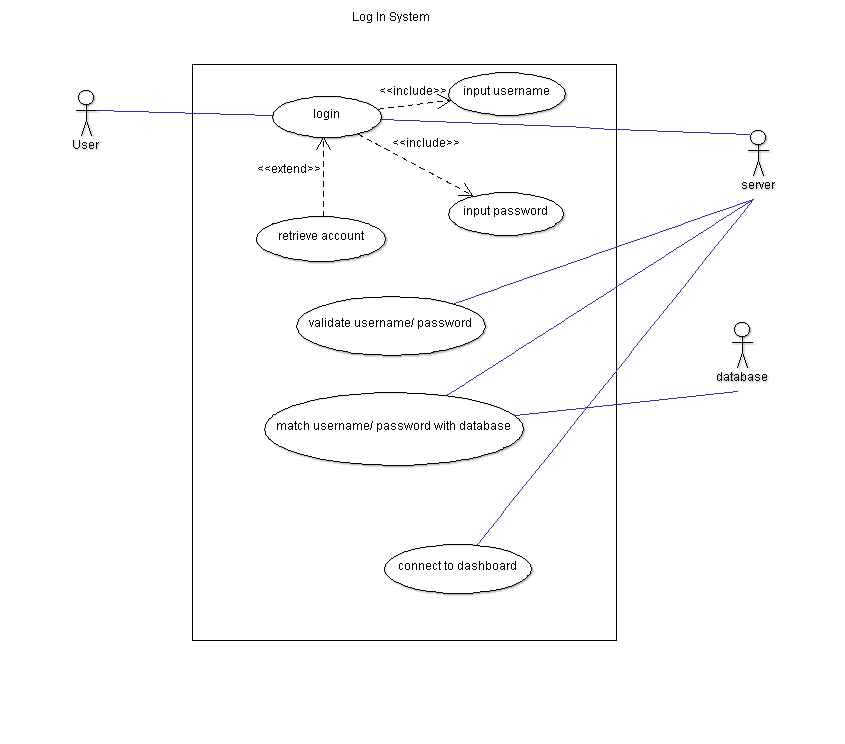
REQ-7: The system shall display an error message if the user input a wrong password

REQ-8: The system shall ask if the user wants to retrieve an account after displaying the login error message

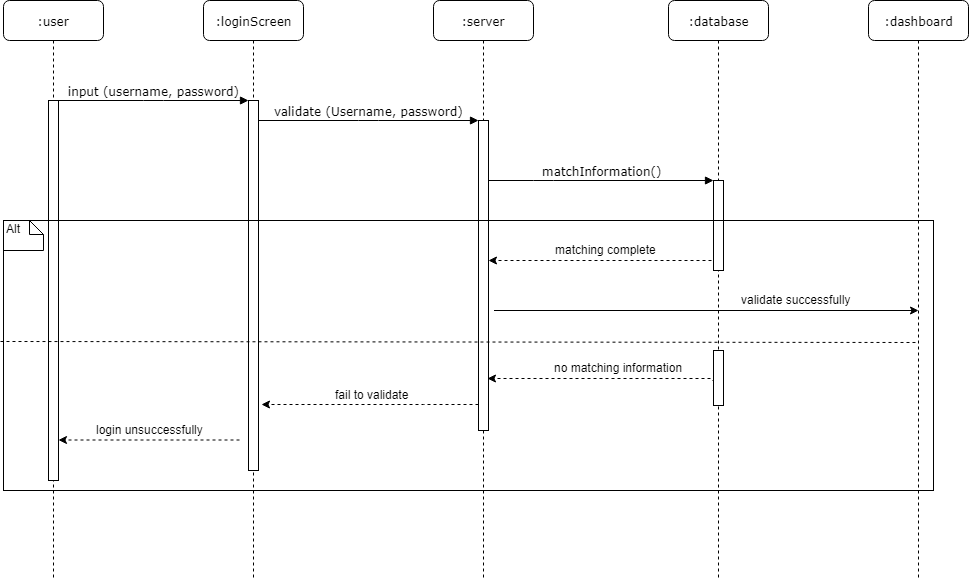
|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 2.2.5 | | |
| Use Case Name: | Login Feature | | |
| Created By: | Linh Huynh | Last Updated By: | Linh Huynh |
| Date Created: | 04-10-2019 | Date Last Updated: | 04-10-2019 |

|  |  |
| --- | --- |
| Actors: | All users |
| Description: | User logs in to melodious system using a username and password |
| Trigger: | User want to have access to free services of the Melodious system |
| Preconditions: | The user is in the log in page. |
| Postconditions: | The system directs user to the dashboard |
| Normal Flow: | 1. User inputs a username and password 2. The system validates the username and password 3. The system matches the username and password with the database [Alternate Scenario A] 4. The system connects user to the dashboard after successfully matching username and password with the database   REQ-6: The system shall display an error message if the user input a wrong username  REQ-7: The system shall display an error message if the user input a wrong password  REQ-8: The system shall ask if the user wants to retrieve an account after displaying the login error message |
| Alternative Flows: | Alternate scenario A - The username or password does not match with the database  A.1 The system display “Your username or password is incorrect”  A.2 The user is returned to 1 |
| Exceptions: | None |
| Includes: | None |
| Priority: | Medium |
| Frequency of Use: | Every time user wants to have access to free services of the Melodious system |
| Business Rules: | 1. The user must be older than 13 years. 2. Password is encrypted |
| Special Requirements: | The user must have an existing account in the Melodious system |
| Assumptions: | None |
| Notes and Issues: | None |

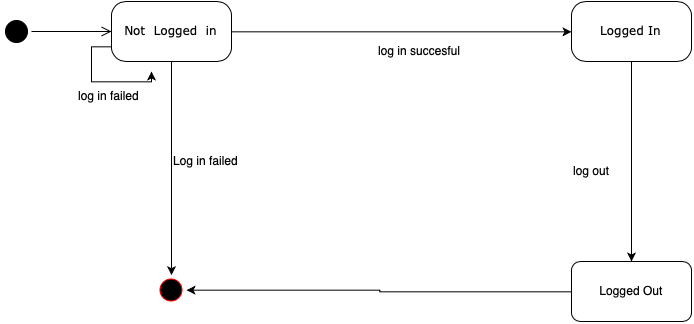
**Table 4.2-1 Use case description of login feature**



**Figure 4.2.a Detailed use case diagram of the login system**



**Figure 4.2.b Sequence diagram of the login system**



**Figure 4.2.c State diagram for the login system**

## 4.3 User Registration Features

**4.3.1 Description and Priority**

The Melodious system allows user to register to the system. These users are registered as regular users. The system does not allow regular users to create events. It is a High priority.

**4.3.2 Stimulus/Response Sequences**

The user submits the registration form.

**4.3.3 Function Requirements**

REQ-1: The system shall display user registration page username, email, date of birth, password fields.

REQ-2: The user shall be able to submit the registration form.

REQ-3: The system shall check the email format.

REQ-3: The system shall verify if the username already exists.

REQ-4: The system shall verify the password follows the rules.

REQ-5: The system shall send the registration confirmation email to the user’s email.

REQ-6: The system shall inform user that the registration confirmation email has been sent to the provided email in the registration.

REQ-7: The user shall be able to view the registration confirmation email.

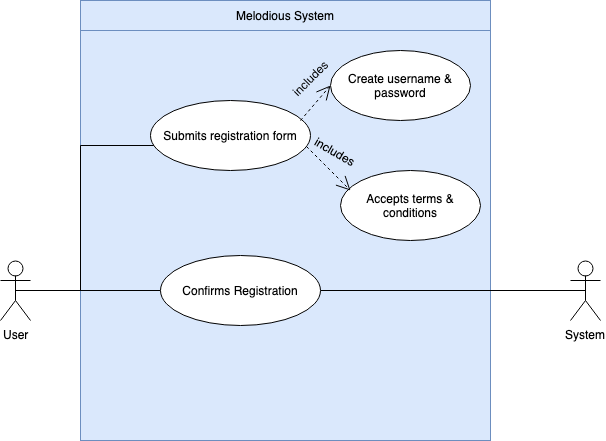
REQ-8: The user shall be able to confirm the registration.

REQ-9: The system shall register the user.

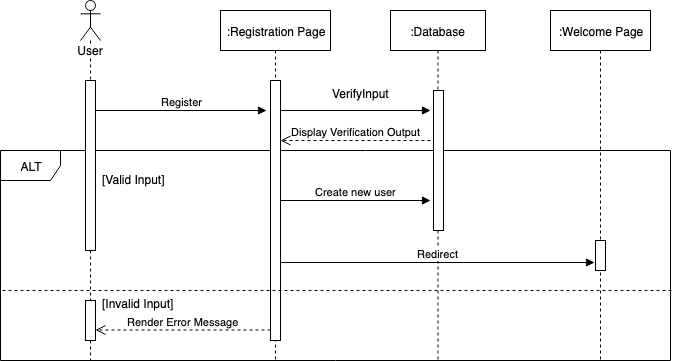
|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 2.2.13 | | |
| Use Case Name: | Regular User Registration | | |
| Created By: | Camy Chhetri | Last Updated By: | Camy Chhetri |
| Date Created: | 04-08-2019 | Date Last Updated: | 04-10-2019 |

|  |  |
| --- | --- |
| Actors: | Regular user, System |
| Description: | User register to melodious system using their name, birthdate, and email. |
| Trigger: | User wants to register to access the Melodious features. |
| Preconditions: | 1. The user is in the registration page. |
| Postconditions: | 1. The system approves registration request. |
| Normal Flow: | 1. Student fills out registration form. 2. Student creates username and password. 3. Student accepts the terms and conditions. 4. Student submits registration form. 5. System verifies user’s information. 6. System confirms registration. 7. System sends a verification email. 8. User clicks on the email. 9. System registers user in the system. |
| Alternative Flows: | 7. The email is invalid.   * 7a1. The user does not get the email confirmation. * 7a2. The user tries again. |
| Exceptions: | The email format is not valid. |
| Includes: | 1. Create username and password 2. Accepts terms & conditions |
| Priority: | High |
| Frequency of Use: | Every time user wants to register to access the Melodious features. |
| Business Rules: | 1. The user must be older than 13 years. 2. Password is encrypted |
| Special Requirements: | None |
| Assumptions: | None |
| Notes and Issues: | None |

**Table 4.3-1 Use case description of the regular user registration**



**Figure 4.3.a Detailed use case diagram of regular user registration**



**Figure 4.3.b Sequence diagram of regular user registration**

## 4.4 Prioritization Algorithm

Since selling concert tickets is one of the most important features for the Melodious system, ticket selling platform which includes points and rewards, third-party payment system, and event lists from Eventful and Ticketmaster has the highest priority. It must be developed first and tested early because of the complex integration with third-party systems.

User registration and Artist registration are prioritized as the second priority and will be developed after ⅓ of the selling ticket platform is completed. Since the registration forms for regular users and artists are similar, both user registration and artist registration have the same priority and can be built concurrently. Login feature has the third priority and must be developed after the regular registration feature is complete. Then, developers can involve users to test the ticket selling platform during the development process and deliver an early version of the product to the stakeholders.

Melodious social media platform has the fourth priority because it is one of the main features that distinguishes Melodious system with competitors in the industry. After ⅔ of the ticket selling platform is completed, software engineers must start adding this feature into the system. Users must be involved during the development process to test the user interface of the system. Other features such as concert video have the lowest priority and may be added at the end or in the incremental versions.

Ticket selling feature, two types of registrations and login feature must be completed first and foremost because these are the basic features for the organization to generate profits through the Melodious system.

# **5.** **Other Nonfunctional Requirements**

## 5.1 Performance Requirements

5.1.1 Melodious system shall create a user account within 1 minute of a user completing the sign up form.

5.1.2 Melodious system shall send the user a confirmation email about their account being created within 1 minute of a user completing the sign up form.

5.1.3 Prior to requesting to become friends with another user, the Melodious system shall display a red outline around the user’s profile picture.

5.1.4 After requesting to become friends with another user, the Melodious system shall display a yellow outline around the requested user’s profile picture within 1 second of their request.

5.1.5 Once the friend request has been accepted by the other user, the Melodious system shall display a green outline around the user’s profile picture within 1 second of their acceptance.

5.1.6 When purchasing ticket(s), the Melodious system shall place the specified number of tickets on hold for the user for 10 minutes.

5.1.6.1 If the user completes the ticket purchase form within the 10 minutes then the Melodious System shall send an email confirmation containing the user’s ticket(s) to the user within 5 minutes

5.1.6.2 If the user does not complete the ticket purchase form within the 10 minutes then the Melodious System shall display the following message on the user interface “Session has timed out” and will take the specified number of tickets off hold for the user.

5.1.7 For iOS users, upon receiving confirmation of a ticket purchase the Melodious System shall allow the user’s ticket(s) to be added to their Apple Wallet.

5.1.8 Melodious System shall be able to support 1,000 concurrent users.

## 5.2 Safety Requirements

Because the Melodious system is a website and a mobile application, there will be no possible harm or damage that could result from the use of the system.

## 5.3 Security Requirements

5.3.1 Melodious transactions shall be completed using HTTPS SSL encryption.

5.3.2 Melodious accounts shall require user login credentials (i.e., username and password) to access account information and profile.

5.3.3 The Melodious system will follow security protocols established within Melodious security policy.

5.3.4 User access permissions for the Melodious System shall only be changed by the system administrators.

5.3.5 Melodious data shall be backed up every 24 hours and secured by using cloud storage.

5.3.6 The Melodious system shall encrypt all communications between users.

5.3.7 The Melodious system shall enforce users to change their passwords every 120 days.

5.3.8 Melodious administrative accounts shall require two authentication methods, password and a personal identification number (PIN).

## 5.4 Software Quality Attributes

5.4.1 If tickets are sold-out or the user is requesting to buy more tickets than there are left, the Melodious. system shall display a negative confirmation message within 1 second of attempting to purchase tickets.

5.4.2 Melodious website and mobile application client side shall be developed using JavaScript.

5.4.3 Melodious shall be developed using an Microsoft Azure SQL database.

## 5.5 Business Rules

5.5.1 Melodious users shall be able to read their account details.

5.5.2 Melodious users shall have access to read and change their account preferences.

5.5.3 Melodious users shall have access to read and write access to their profile details.

5.5.4 Melodious users shall have access to purchase up to 10 tickets.

5.5.5 Melodious users shall have access to add, delete, or block other users.

5.5.6 Melodious administrators shall have add, delete, and restrict privileges on user accounts.

5.5.7 Melodious administrators shall have privileges to approve or deny artist verification forms.

5.5.8 Melodious administrators shall be able to add, update, or delete concert details.

5.5.9 Melodious user accounts shall be created when users sign up on the Melodious website.

# **6.** **Other Requirements**

## 6.1 Third-party system requirements

6.1.1 Third-party payment system shall include PayPal, Visa, Discover, and Mastercard