AUTOTEAU

QuickKar

Software Requirements Specification

<2.0>

Revision History

Date	Version	Description	Author

Contents

Software Requirements Specification

1. Introduction

1.1 Purpose

This document provides a high-level detailed description of QuickKar system. It includes the explanation of system's feature, the flow process, the design interface of software and how system deal with error and unexpected situation. This SRS document is intended for both stakeholders and project team.

1.2 Scope

QuickKar is a Java desktop program that maintains a list of song for local karaoke business in HCMC. This software will be designed to make a comfortable for both customer and karaoke's staffs. By minimum the unnecessary features but do not lose the requirement's needs, the system will work faster, more effective and easy to use. The system is designed for 3 type of user: customer, karaoke's staffs and admin.

1.3 Assumptions and Dependencies

1.4 References

- http://www.agilemodeling.com/essays/agileModelingRUP.htm
- Priestley, M. (2004). *Practical Object-Oriented Design with UML.* Mc Graw Hill.

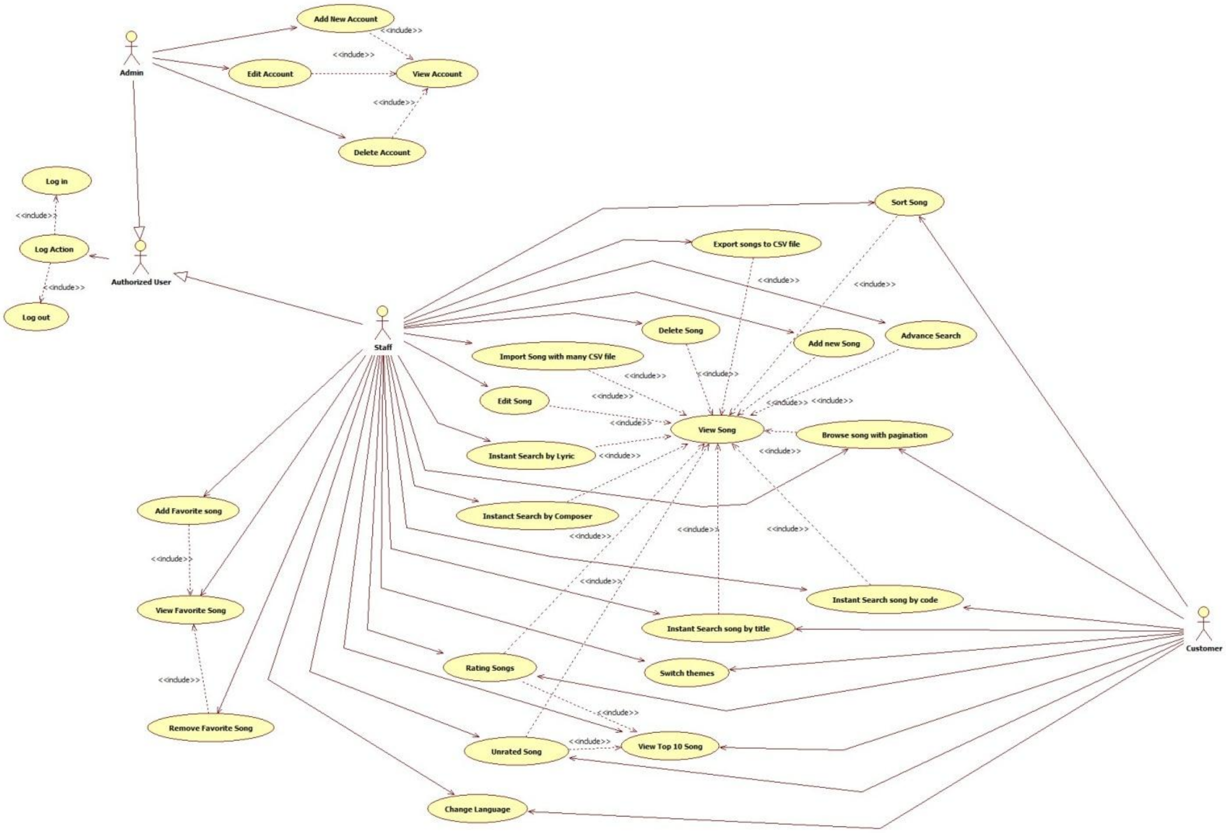
2. Specific Requirements

- **QuickKar** is the application that maintains a list of songs for a local karaoke in HCM city. Thus, it provides adding, editing, deleting song ability and importing a CSV-song file to the database. Moreover, end-users can search for the song they want through song's code or song's title or change between two different GUI of the program.

2.1 Actors

The application will contain three kinds of account which are admin, staff and customer.
+Admin account is the only one administration account which has the constant username and editable password. The username and password of this type of account is the same which is “admin”. This administrator will primary manage types of account in this program such as: adding new staff, viewing and editing staff’s information and delete staff’s account.
+ Staff account which is created by admin. The fundamental responsibility of staff is adding new song, viewing or editing song’s information, deleting song and importing songs from multiple CSV files. Staff can also have all functions of customer account.
+Customer account doesn’t need to log in to the system in order to use it. Customer can browse songs with paginations, instant search or advance search song by code, title, lyric and composer, Moreover, customer can vote for the top ten 10 songs, add favorite songs to the favorite list and be able to switch between two different themes.

2.2 Use Case Diagram



2.3 Use Case Specifications

2.3.1 *Login*

Use Case Name: User log in	ID: UC001
Actors: Authorized user	Priority: High
Brief description: Authenticate the user	
Pre-conditions: Account must have been created	
Trigger: Validate user	
Post-conditions: User must log out to ensure the security for the system	
Normal flow: +User types user name, password, and then proceed to next step by typing "Enter" or clicking on "Login" Button	
Exceptional flows: +If wrong user name or password, a warning popup show up and ask user to re-type username and pass word. +If the username and password fill boxes are empty, system will ask user to enter username and password.	

2.3.2 *Logout*

Use Case Name: User log out	ID: UC002
Actors: Authorized user	Priority: Normal
Brief description: Log out	
Pre-conditions: User must log in first	
Trigger: User finishes using the system	
Post-conditions: Logged out	
Normal flow: User enter user name, password, and then proceed to next step by typing "Login" Button. User can get back to choosing user page by click "Back" button.	
Exceptional flows:	

2.3.3 *Add new Account*

Use Case Name: Add new account	ID: UC003
Actors: Admin	Priority: Normal
Brief description: Add account staff information	
Pre-conditions: Admin must log in, the created account is not duplicated with the existing account on the system	
Trigger: The new staff has been employed	
Post-conditions: Admin logs out after finishing	
Normal flow: +Admin must press the icon of adding new User, and then the popup will occur. +Admin can fill in username, password and other information fields. +Press "Enter" or button "OK" to finish the action of adding new Account	
Exceptional flows: A warning frame show up if there exist invalid format information or the account that is going to create has been existed in the system.	

2.3.4 *Edit Account*

Use Case Name: Edit account	ID: UC004
Actors: Admin	Priority: 5
Brief description: Edit information of staff or admin	
Pre-conditions: Admin must login; Accounts must be created first	
Trigger: Admin wants to change staff information	
Post-conditions: Admin logs out; Information will be changed	
Normal flow: +Admin choose the account that he want to edit information. +Admin clicks on Edit button, and then fill in or change necessary information +Press "Enter" or Button "OK" to finish the action	
Exceptional flows: If the information changed format is not valid, a warning pop up and tell admin where the errors are.	

2.3.5 *Delete Account*

Use Case Name: Delete account	ID: UC005
Actors: Admin	Priority: Normal
Brief description: Delete account	
Pre-conditions: Admin must log in first;	
Trigger: Admin wants to delete unused account	
Post-conditions: Admin can view existing accounts or log out	
Normal flow: + Admin choose the accounts that he want to delete. +Admin chooses accounts and confirm on delete.	
Exceptional flows: The admin accounts cannot be delete because it is administrator account	

2.3.6 *View Account*

Use Case Name: View account	ID: UC006
Actors: Admin	Priority: Normal
Brief description: Admin can view all available accounts in the system	
Pre-conditions: Admin must login	
Trigger: After admin log in	
Post-conditions: Admin can add, edit or delete account	
Normal flow: Admin views available account with full information directly after logging to the system.	
Exceptional flows: There is no available account in the system except admin account	

2.3.7 *Add New Song*

Use Case Name: Add song	ID: UC007
Actors: Staff	Priority: 5
Brief description: Adding song	
Pre-conditions: Staff must log in	
Trigger: Staff wants to add new song to the list	
Post-conditions: Staff can view the song list	
Normal flow: +Staff clicks on the button “Add Song”, and then the system displays a form for staff to input the information of song. +After finishing adding information, Staff can press “Enter” or button “OK” to finish the action of adding new Song	
Exceptional flows: A warning pop up if the song that is entered has already been existed in the system or invalid format song code.	

2.3.8 *Delete Song*

Use Case Name: Delete song	ID: UC008
Actors: Staff	Priority: 5
Brief description: Delete song	
Pre-conditions: Staff must log in	
Trigger: Staff wants to delete songs from the list	
Post-conditions: Staff can view the song list after deleting	
Normal flow: Staff clicks on the “Delete button”, and then the system displays list of songs with the check box next to, user clicks on the check box to choose the song he or she wants to delete. Finally, click on the confirm button to delete the song	
Exceptional flows: Staff has not chosen any song to delete, system do nothing.	

2.3.9 *Import Song from CSV file*

Use Case Name: Import songs	ID: UC009
Actors: Staff	Priority: 5
Brief description: Import song from the pdf or csv file	
Pre-conditions: Staff must log in	
Trigger: Staff wants to import list of songs from existing pdf or csv file	
Post-conditions: Staff can view the song list that is added	
Normal flow: +Staff clicks on the import file; +Then the system opens a window for user to choose multiple csv files that staff wants to import. +Finally, staff clicks on the "Open" button. The system opens log window display what are done	
Exceptional flows: The song cannot import to the song list	

2.3.10 *Edit Song*

Use Case Name: Edit song	ID: UC010
Actors: Staff	Priority: 5
Brief description: Edit song information	
Pre-conditions: Staff must log in the systems	
Trigger: Staff wants to edit songs	
Post-conditions: Staff can view the song list that has been edited	
Normal flow: +Staff clicks on the song and clicks on the "Edit Button" + The form for editing will occur and staff can edit the information of songs. +Finally, press "Enter" or button "OK" to finish the action of editing song.	
Exceptional flows: The information of song being edited is not valid (for example, the user account is already existed in the system)	

2.3.11 *View Song*

Use Case Name: View songs	ID: UC011
Actors: Staff or Customer	Priority: 5
Brief description: Allow user to view list of song	
Pre-conditions: +With Staff: must log in to the system. +With Customer: need run the system only.	
Trigger: User wants to view the list of songs	
Post-conditions: User can view the song with pagination	
Normal flow: User clicks on page number to browse between song list	
Exceptional flows: The list does not have any song	

2.3.12 *Instant Search by Title*

Use Case Name: Instant Search song by title	ID: UC012
Actors: Staff or Customer	Priority: 5
Brief description: Allow users to search song by title	
Pre-conditions: +With Staff: must log in to the system. +With Customer: need run the system only.	
Trigger: Users wants to search song by title	
Post-conditions: Users can view the song list with the given words	
Normal flow: + User enters the name of the song in the search field + Then the system will displays a list of song that is likely to have that keyword in the song name automatically. The found word will be bold.	
Exceptional flows: The song name entered is not in the system	

2.3.13 *Instant Search by Code*

Use Case Name: Instant Search song by code	ID: UC013
Actors: Staff or Customer	Priority: 5
Brief description: Allow users to search song by code	
Pre-conditions: +With Staff: must log in to the system. +With Customer: need run the system only.	
Trigger: Users wants to search song by code	
Post-conditions: Users can view the song list with the given code	
Normal flow: + User enters the code of the song in the search field + Then the system will displays a list of song that is likely to have that keyword in the song code automatically. The found word will be bold.	
Exceptional flows: The code entered is not in the system	

2.3.14 *Instant Search by Composer*

Use Case Name: Instant Search song by Composer	ID: UC014
Actors: Staff or Customer	Priority: 5
Brief description: Allow users to search song by code	
Pre-conditions: +With Staff: must log in to the system. +With Customer: need run the system only.	
Trigger: Users wants to search song by code	
Post-conditions: Users can view the song list with the given composer	
Normal flow: + User enters the composer of the song in the search field + Then the system will displays a list of song that is likely to have that keyword in the song composer automatically. The found word will be bold.	
Exceptional flows: The composer entered is not in the system	

2.3.15 *Instant Search by Lyric*

Use Case Name: Instant Search song by Lyric	ID: UC015
Actors: Staff or Customer	Priority: 5
Brief description: Allow users to search song by code	
Pre-conditions: +With Staff: must log in to the system. +With Customer: need run the system only.	
Trigger: Users wants to search song by code	
Post-conditions: Users can view the song list with the given lyric	
Normal flow: + User enters the Lyric of the song in the search field + Then the system will displays a list of song that is likely to have that keyword in the song lyric automatically. The found word will be bold.	
Exceptional flows: The lyric entered is not in the system	

2.3.16 *Switch Theme*

Use Case Name: Switch theme	ID: UC016
Actors: Staff or Customer	Priority: 5
Brief description: Allow users to switch GUI appearance	
Pre-conditions: +With Staff: must log in to the system. +With Customer: need run the system only.	
Trigger: Staff wants to change the theme of the software	
Post-conditions: Staff can view different theme of software	
Normal flow: User clicks on View, and then change theme	
Exceptional flows: The system is being processed, so cannot change to other themes	

2.3.17 *Advance Search*

Use Case Name: Advance Search	ID: UC017
Actors: Staff or Customer	Priority: 5
Brief description: Allow users to search the exact word or phrase inside a pair of double quotes. E.g. “con duong”	
Pre-conditions: +With Staff: must log in to the system. +With Customer: need run the system only.	
Trigger: Users wants to use advance search.	
Post-conditions User can view available song list of software	
Normal flow: +User can enter the word that he wants to search and put it in the pair of double quotes. + Then the system will displays a list of song that is likely to have that keyword in the song automatically. The found word will be bold.	
Exceptional flows: The word or phrase that is entered not exists in the system.	

2.3.18 *View Top 10 song*

Use Case Name: View Top 10 song	ID: UC018
Actors: Staff or Customer	Priority: 5
Brief description: Allow users can view the list of top 10 songs that are rated. The list is sorted from the first to the tenth.	
Pre-conditions: +With Staff: must log in to the system. +With Customer: need run the system only.	
Trigger: Users wants to view the list of top 10 rated songs.	
Post-conditions User can view available song list of software	
Normal flow: + User can see the list of top 10 voted song directly in the right part of the main view after logging in to the system.	
Exceptional flows: +There is no song that rated. Therefor the list will be empty. + The list only shows the number of song that is rated, and the number of song will be up to 10.	

2.3.19 *Rate Song*

Use Case Name: Rate song	ID: UC019
Actors: Staff or Customer	Priority: 5
Brief description: Allow users to rate the song. Each user can vote a song one time. However, they can change the final result any time. E.g.: first they vote 2 starts for that song, and then they can change their decision by vote 5 starts. Thus, the final result voting for that song of user is 5 starts.	
Pre-conditions: +With Staff: must log in to the system. +With Customer: need run the system only.	
Trigger: Users wants to rate the song that they love.	
Post-conditions User can view available song list of software	
Normal flow: +User can click on the song that they want to rate. + In the rate column, user can choose how many stars they want to vote for that song by click on the row of stars.	
Exceptional flows: +There is no song for voting in the system	

2.3.20 *Add Favorite Song*

Use Case Name: Add Favorite Song	ID: UC020
Actors: Staff or Customer	Priority: 5
Brief description: Allow users to add the songs to the favorite list song. It helps them to manage their favorite songs.	
Pre-conditions: +With Staff: must log in to the system. +With Customer: need run the system only and enter their phone numbers.	
Trigger: Users wants to add the song that they love to the favorite list.	
Post-conditions User can view available song list of software	
Normal flow: +User can click on the song that they want to add. + Then choose the button "Add Favorite" or right click and choose "Add favorite" to finish the action.	
Exceptional flows: +There is no song for adding in the system + The song is added has been already existed in the favorite list of that user.	

2.3.21 *View Favorite List*

Use Case Name: View Favorite Song	ID: UC021
Actors: Staff or Customer	Priority: 5
Brief description: Allow users to view the favorite list song that they create before.	
Pre-conditions: +With Staff: must log in to the system. +With Customer: need run the system only and enter their phone numbers.	
Trigger: Users wants to add the song that they love to the favorite list.	
Post-conditions User can view available song list of software	
Normal flow: +User can click on the icon "Favorite Song" + Then the form will occur and display all the favorite songs that are added to the list.	
Exceptional flows: +There is no song in the favorite list.	

2.3.22 *Remove Favorite song*

Use Case Name: Delete Favorite Song	ID: UC022
Actors: Staff or Customer	Priority: 5
Brief description: Allow users to remove the favorite song from the favorite list.	
Pre-conditions: +With Staff: must log in to the system. +With Customer: need run the system only and enter their phone numbers. + The song that user want to remove must exist in the favorite list	
Trigger: Users wants to remove the song that they love to the favorite list.	
Post-conditions User can view available song list of software.	
Normal flow: +User can click on the icon "Favorite Song" + Then the form will occur and display all the favorite song. + Click on the button "Remove" to remove the favorite song. +Then the action will remove that song from the favorite list.	
Exceptional flows: +There is no song in the favorite list.	

2.3.23 *Unrated Song*

Use Case Name: Unrated song	ID: UC023
Actors: Staff or Customer	Priority: 5
Brief description: Allow users to un-rate the song that they have already voted before.	
Pre-conditions: +With Staff: must log in to the system. +With Customer: need run the system only.	
Trigger: Users wants to un-rate the song that they want.	
Post-conditions User can view available song list of software	
Normal flow: +User can click on the song that they want to rate. + Click on the rate column, user can choose 0 stars they want to remove rating for that song.	
Exceptional flows: +There is no song for voting in the system	

2.3.24 *Sort Song*

Use Case Name: Sort Song	ID: UC023
Actors: Staff and Customer	Priority: 5
Brief description: Allow user to sort songs by code, title, author, lyrics and rating.	
Pre-conditions: +With Staff: must log in to the system. +With Customer: need run the system only.	
Trigger: users want to sort songs.	
Post-conditions: User can view available song list of software	
Normal flow: + Users click the sort song check box to sort song by code, title, author, lyrics, rating.	
Exceptional flows:	

2.3.25 *Change Language*

Use Case Name: Change Language	ID: UC023
Actors: Staff and Customer	Priority: 5
Brief description: Allow user to the language to another.	
Pre-conditions: +With Staff: must log in to the system. +With Customer: need run the system only.	
Trigger: users want to change language.	
Post-conditions: User can view available song list of software	
Normal flow: + User click on the button which show another language to change to another	
Exceptional flows:	

2.3.26 *Export songs to CSV file*

Use Case Name: Export songs to CSV file	ID: UC023
Actors: Staff	Priority: 5
Brief description: Allow user to export songs to CSV file.	
Pre-conditions: +With Staff: must log in to the system.	
Trigger: users want to export songs to CSV file.	
Post-conditions: User can view available song list of software	
Normal flow: + click the Export button to export songs to CSV file.	
Exceptional flows: There is nothing to export songs	

2.4 Non-Functional Requirements

2.4.1 *Installability*

- **QuickKar** will be written as a java desktop program which can operate on multiple platform such as Window (2000/XP/7) or Linux (Fedora 14+ or Ubuntu 10/04+). Moreover, the minimum hardware for this application will be the Pentium 3 with 1GB RAM, 521MB free hard disk space.
- **QuickKar** will be provided as the executable jar file named “QuickKar2012-Autoteau” instead of the source code. Thus, after installing the jar file to the computer, user can run the application directly through the icon logo of the program.

2.4.2 *Usability*

AUTOTEAU will provide the easy understanding interactive interface that every user can adapt the **QuickKar** immediately. Moreover, there is a User Guide that can help users clearly about show to use each function in the program.

2.4.3 *Performance*

- When the program firstly starts, it will take about 4-5 second to load the database of songs and accounts. It depends on the amount of database which leads to the short or long response time Every time the users press the every button; it will take mostly 1 second to do the action. Nevertheless, with the search function, whenever users finish typing the keyword in the search field, time to find the song is quite quick which is often less than 1 second automatically. Furthermore, advance search also the important feature of the program whether user can type the keyword inside a pair of double quotes and the result will also display mostly less than 1 second with the bold word in the song code, lyric, title and composer which look like the keyword in the search field. Finally, when the users want to turn off the system, the system will auto save all data to the database. This will take about mostly 1 second.

- Importing song with totally 45 karaoke volumes which contain for than 5000 songs will take about 10 seconds for the application to load all data. Depending on the capacity of the computer, the time will take shorter or longer. Moreover, when users want to search a song, the results will be showed no more than 1 second.
- About the security, the users must log in to the system in order to modify accounts as the admin, or modify songs as the staff of the system.

2.4.4 *Scalability*

- **QuickKar** can be imported thousands songs with multiple languages such as English, Vietnamese, etc. In the future, this application will be added more functions which can satisfy customer's need about the karaoke like connecting to the server for listening online music. Nevertheless, the more song is added to the system, the huge database is increased.
- The application will have the GUI that fit with the optimized resolution of 1024 x 768 pixels.

2.4.5 *Technical*

- During writing code for the application, some other methodologies and tools which are used to support for the works such as RUP which is used for the application document, Unfuddle which can help **AUTOTEAU** to manage the source code in the server in order to prevent any accidentally actions happened such as losing files, error files.
- The application also contains the JUnit test in the Test package which tests all classes in the Model package.