

COMMUNITY COLLEGE OF CITY UNIVERSITY

DIVISION OF APPLIED SCIENCE AND TECHNOLOGY

AST 20613 Computer System Development

2014-2015

V. Calendar

(Voice Calendar)

FINAL YEAR PROJECT

SYSTEM SPECIFICATION

by

Group:ISD02

WONG, Ho Long(53587834)\*

LI, Sing Lun (53612386)

SIN, Sin Chung(53587350)

WONG, Chun Kit(53581516)

Supervised by

Dr. Timothy Kwong

# **ACKNOWLEDGEMENTS**

We have taken efforts in this project, it would not have been possible without the kind help and support of many individuals and organizations. We would like to take this chance to extend our sincere thanks of all of them.

We are very thankful to our supervisor, Dr.Timothy Kwong, the guide of the project that helps us to solve many problems throughout the project. Also, we are highly indebted for his guidance and constant supervision as well as for providing necessary information regarding the project and also for his support in completing the project. Moreover, his kind concern provides much power to our team to finish the project successfully.

We would also like to express our special gratitude and thanks to the teachers who have taught us in these two years. The knowledge they have taught us is very useful in this project and it helps to fix lots of issues.

The teachers that we owe a great many thanks are: (Names are listed in alphabetical order)

Dr. Desmond Tsoi

Mr. Frank Lau

Mr. Jackie Kwong

Dr. Kenny Ma

Dr. Timothy Kwong

We would also like to thank you all of our groupmates, with all the help and support for doing the development of the project.

Thank you.

Yours sincerely,

Wong Ho Long, Li Sing Lun, Sin Sin Chung, Wong Chun Kit

FYP ISD-02 V. Calendar

Table of Contents

Acknowledgements

Abstract

1. INTRODUCTION

* 1. BACKGROUND
  2. PROJECT OVERVIEW

1.3 OBJECTIVE

1.4 PROJECT SCOPE

2. SYSTEM REQUIREMENTS

2.1 FUNCTIONAL AND DATA REQUIREMENTS

2.2 DATA ELEMENTS

2.3 USE CASE MODEL

2.4 DATA STORE

2.5 PROCESS DESCRIPTION

2.6 EXTERNAL ENTITIES DESCRIPTION

3. DATABASE DESIGN

3.1 DATA ANALYSIS

3.2 TABLE DEFINITION AND DESCRIPTION

3.3 DATA DESCRIPTION

3.4 OVERALL ATTRIBUTES OF VOICE CALENDAR

4. SYSTEM DESIGN

4.1 CALENDAR

4.2 CREATE EVENT

4.3 CREATE AND EDIT DETAILS OF EVENT

4.4 DAY SCHEDULE

4.5 MONTH SCHEDULE

5. TEST PLAN

5.1 OBJECTIVE

5.2 TESTING PLAN

5.3 TESTING SCOPE

5.4 TESTING STRATEGY

5.5 MODULE-BASED TESTING

5.6 CASE-BASED TESTING

5.7 PASS OR FAIL CRITERIA

5.8 SUSPENSION CRITERIA

5.9 TESTING SCHEDULE

6. LIMITATIONS & FURTHER DEVELOPMENTABSTRACT

The population of portable device is growing gradually especially throughout these year, portable device may become replaced position of Computer or laptops, Not only the student studied in IT field, but also other streams of students usually use portable device in their daily life. Therefore our aim of our project is developing tools for useful managing their life by portable device friendly. Perhaps our project will try to use CSS3 and HTML5 for designing the layout of website, CSS3 and HTML5 are the too use for development.

These tools most update function included flexible property and also animation function, slider function etc, with database application for development such as MySQL. Our group is confident that our development tools will cater all requirements of portable device users indeed.

By ISD02 in 2014-2015

1.1. Background

Nowadays, we can see lots of Hong Kong people using their smart devices such as smart phones, tablets at any time. With the great popularity of smart products, those devices have become part of their live and they cannot live without it. The aim of those smart products is to facilitate people live and make their life become convenient. People can finish their jobs or obtain the specific information easily in a short period of time by using those smart devices and so the number of people using smart products is increasing sharply in recent years.

Besides, why those smart devices can improve people’s life or get the job done easily? First of all, smart devices are portable. People can bring it outside and use it in any places with a light afford, they can put it in their bags or just take it by hand. With the compare of traditional desktop computer or notebook, smart devices are smaller in size and lighter in weight. Also, the design of smart devices fits human’s hand so people can hold it easily and use it for a long time. Secondly, with the aid of physical size of portable devices, people can obtain information or complete a specific job easily at outdoor. For example, people can search the nearby restaurants, the review of the restaurants, the address of shops or sending emails at anywhere by using smart devices. People can obtain what they want in a simple gesture of finger. Thirdly, the functions provided by apps in smart devices can bring a bunch of useful tools that can get the job done easily. Usually, people can use web browser to do specific task and search the things they want by different web pages.

However, although there is a lot of web pages about calendar to help them to notify them what they need to do on that day, they always need to input the details of an incident by their hands. All calendars do not provide a voice input for users to input the details and voice output for users to listen the reminder of that day from the smart devices. It is difficult for those visually impaired people to use those calendars. With the issue mentioned above, we would like to complete some of the functions missed in a specific webpage to make the calendar more perfect. The details of the project will be discussed in the later section.

1.2 Project Overview

Voice Calendar consists of two parts which are webpage side and server side. The brief descriptions of the functions of two parts are listed follow.

WEBPAGE SIDE

1. Voice Input

Users can use their voice to input the details of the event and input the command to the server. The server will implement the relevant command to do what the users want to do.

Whatmores,the user could input their information according to voice guideline,but also the system could automatics for detect whether the user input is correct,if the input is not match the standard(such as detect wrong),the system could required users to input against.The Voice Input system are also smart to detect the string into numbers(such as detect January to 01 as well as month input)

2. Create Event

Users can create their own event to the calendar. After creating an event, users can use the calendar to remind them when do they have things to do today or not.

3. Edit Event

User can edit their information by just clicking their own time slot as well as for modify their data and detailed information.

4. Delete Event

User can delete their event by clicking their own time slot,and pop out message may be announced when the users request for delete timeslot functions.

5.Voice Control

The user could used their commands for control the system for functioning,as well as for go to other page.

SERVER SIDE

1. Login Procedure

Users will be asked to use voice to input their username and password to login the system. Server used to authenticate users to check they are new users or registered before. If they are new users, there are no related information in the database and server will ask users to register an account first. If they are registered before, they can directly login to the system.

2. Voice Output

Server will generate the script that the system need to speak out. The system will according to the page of users searching and send the instruction to server to generate the script. After generating the script, the server will pass it to the system and the system will give a voice output for users to listen what they need to do next.

ADMIN SIDE

1.Admin the accounts editing and delete the accounts

The registered administrators has right for editing the account information and email,while they couldn ‘t have right for administrate the password as privacy issues,the editing function have maintained with high consistency.

1.3 Objective

In general, we would like to focus on making a webpage called Voice Calendar. The project would introduce Voice Recognition engine to obtain users’ input and output the information from the calendar. We hope our target users can use Voice Calendar to help them provide reminder to remember the things by using voice input and output.

In general, this project has three objectives and they are listed below.

1. To facilitate the use for visually-impaired people.
2. To provide a better function based on Google calendar
3. A more user-friendly and configurable UI should be used.

We have seen that the calendar provided by Google cannot satisfy some of the needs of users in daily life. In general, we see that people will face the problems on viewing the calendar when their hands are not “free”. The Voice Recognition engine can enhance the situation they meet. Also, the system can provide a more convenient way to check the schedule on users’ calendar. In the side of UI design, we should provide more settings to users to configure the application and make it more user-friendly.

# **1.4 SCOPE**

As we all know, people use the Google Calendar frequently but they need to use their eyes and hands to control it. They cannot use Google Calendar when they are doing others things. It causes inconvenience. For example, people cannot view their calendar during driving or working at the same time.

In order to make the Google Calendar to be more convenient and cater to universal usability, the following listed out some of the features that we aim to include in GV Calendar to solve the problems and inconvenience that those target users encounter.

1.4.1 Voice output system

Using voice commands to input event in users’ schedule, for cater to users with color blindness and the users who are busy.

->Function only implemented with today schedule

->AI could read aloud the detail, schedule date detail, Venue, Remark accordingly.

1.4.1.1 Read the timetable in the current time

The system could automate operated the timetable with current time only, as well as the event start in today could read aloud by the AI system only.

1.4.2 Voice Input Function

The voice command could be used for input the context and information instead of direct hand input, this method could aids the visual problem patients easily input the information besides from the traditional method.

1.4.3 Voice Control Function

The users could control their calendar by voice easily, such as given the command for register the account, the system could provide different guidelines for the users especially true for visual patients for control their calendar by voices.

1.4.4 Setting the Calendar into different dimensions

For favor with different people style, the calendar provides two types of calendars for user chosen which on is suitable for them, which are Day Schedule and Month Schedule.

1.4.5 Responsive Website Design

The website could function responsively, whatever the users are using mobile, laptop computer or Ipad, the users could easily check the information by the system setting and monitors size accordingly in different cases.

1.4.6 Low Memory required

Most users from out targets are mainly mobile users and portable device users, therefore design should be aims at low memory required, hence made the efficiency of the website maximize.

1.4.7 High Security Maintenance

1.4.7.1. Hashed Password

The password for users could hash for increased sense of security as well as for highly protected the security of users and protected their privacy in the highest security ways.

1.4.7.2. Session for storing the datum instead of cookies

System implemented with the session instead of cookies, which are text file for storing the datum memory for login. Compared to cookies, session has a higher security protection of datums and prevents data get loss by spyware.

1.4.8. Administration Function

The administrator could have the right for control and edit the record of accounts for all the users, if the users have problems for their accounts, they might contact their administrators or coordinators.

2.System Requirement

Voice calendar allow diverses of users for access. The performance of the system relies a lot for google server and web usage. This following will divide the application into different separated parts as well as illustrates the processes for the application works.

2.1 Functional and Data Requirements

It is the ability to apply the correct and appropriate system requirements from different stakeholders and specify them in a manner understandable to them so those processes.

Possible usage of use case modeling for:

- Problem analysis

- Requirements analysis

- Logical design

2.2 Data Element

The data element described the usage of different element names and actual meaning respectively.

2.2.1 Register

|  |  |
| --- | --- |
| Data element name | Name |
| Alias |  |
| Pic | VARCHAR(40) |
| Values/Meaning | To store the new register users’ name |

|  |  |
| --- | --- |
| Data element name | Password |
| Alias |  |
| Pic | VARCHAR(40) |
| Values/Meaning | To store the new register users’ password |

|  |  |
| --- | --- |
| Data element name | RepeatPassword |
| Alias |  |
| Pic | VARCHAR(40) |
| Values/Meaning | To confirm users input the same password as before |

2.2.2 Account

|  |  |
| --- | --- |
| Data element name | Name |
| Alias |  |
| Pic | VARCHAR(40) |
| Values/Meaning | To check the username is it on this database for login system |

|  |  |
| --- | --- |
| Data element name | Password |
| Alias |  |
| Pic | VARCHAR(40) |
| Values/Meaning | To check the password is it in this database for login system |

2.2.3 Event

|  |  |
| --- | --- |
| Data element name | Id |
| Alias |  |
| Pic | INTEGER(11) |
| Values/Meaning | To store an id of an event |

|  |  |
| --- | --- |
| Data element name | Subject |
| Alias |  |
| Pic | VARCHAR(1000) |
| Values/Meaning | To store the title of the event |

|  |  |
| --- | --- |
| Data element name | Location |
| Alias |  |
| Pic | VARCHAR(200) |
| Values/Meaning | To store the location of the event |

|  |  |
| --- | --- |
| Data element name | Description |
| Alias |  |
| Pic | VARCHAR(255) |
| Values/Meaning | To store the detail information or reminder of the event |

|  |  |
| --- | --- |
| Data element name | StartTime |
| Alias |  |
| Pic | DATETIME |
| Values/Meaning | To store the starting time of the event |

|  |  |
| --- | --- |
| Data element name | EndTime |
| Alias |  |
| Pic | DATETIME |
| Values/Meaning | To store the ending time of the event |

|  |  |
| --- | --- |
| Data element name | IsAllDayEvent |
| Alias |  |
| Pic | SMALLINT(6) |
| Values/Meaning | To store the event is it full day event |

|  |  |
| --- | --- |
| Data element name | Color |
| Alias |  |
| Pic | VARCHAR(200) |
| Values/Meaning | To store the color of the title of the event |

|  |  |
| --- | --- |
| Data element name | RecurringRule |
| Alias |  |
| Pic | VARCHAR(500) |
| Values/Meaning | To check the event is it more than one time to do |

2.2.4 Admin Register

|  |  |
| --- | --- |
| Data element name | Name |
| Alias |  |
| Pic | VARCHAR(40) |
| Values/Meaning | To store the new admin register name |

|  |  |
| --- | --- |
| Data element name | Password |
| Alias |  |
| Pic | VARCHAR(40) |
| Values/Meaning | To store the new admin register password |

|  |  |
| --- | --- |
| Data element name | RepeatPassword |
| Alias |  |
| Pic | VARCHAR(40) |
| Values/Meaning | To confirm administrators input the same password as before |

2.2.5 Admin Account

|  |  |
| --- | --- |
| Data element name | AdmName |
| Alias |  |
| Pic | VARCHAR(40) |
| Values/Meaning | To check the admin name is it on this database for login system |

|  |  |
| --- | --- |
| Data element name | Password |
| Alias |  |
| Pic | VARCHAR(40) |
| Values/Meaning | To check the admin password is it on this database for login system |

2.3 Data flow Diagram

The data flow diagram could likely to divided as Use Case Glossary,application and Narrative respectively.In fact,the Use case Glossry ‘s usage mainly refers for the action and use case brief description,while Narrative is refer for the Use Case Model and detalized information for each activity.

2.3.1. Use Case Glossary

The Use case glossary have summarized as different activities for different users usage,for the further usage diagram ‘s basic background and introduction.

|  |  |  |
| --- | --- | --- |
| Use-Case Glossary | | |
| Use-Case Name | Use-Case Description | Participating  Actors and Roles |
| Create account | This use case describes the event of creating a new account. | Client |
| Enter account | This use case describes the event of a client entering their account and password. | Client |
| Enter event | This use case describes the event of a client add new event into their calendar | Client |
| Check event | This use case describes the event of a client checking in their calendar | Client |
| Text to speech | This use case describes the event of automatically generate a speech from text. | Time |
| Speech to text | This use case describes the event of automatically transferring speech to text | Time |
| Edit event | This use case describes the event of editing event in the calendar | Client |
| Logout | This use case describes the event of logout user’s account | Client |

2.3.2. Use Case Diagram

2.3.2.1. Terminology of Use Case Diagram:

Term1 : Use case – a subset of the overall system functionality

Represented by a horizontal ellipse with name of use case inside the ellipse

Term 2 : Actor – anyone or anything that needs to interact with the system to exchange information human, organization, another information system, external device, time, etc…



2.3.2.1.Actor Symbol

Term 3 : Association - a relationship between an actor and a use case in which an interaction occurs between them



2.3.2.2.Association Diagram Symbol

2.3.2.2.Use Case Diagram

The Use case diagram clearly illustrated the process of interchange between calendar server and users ‘s flow and activities respectively.

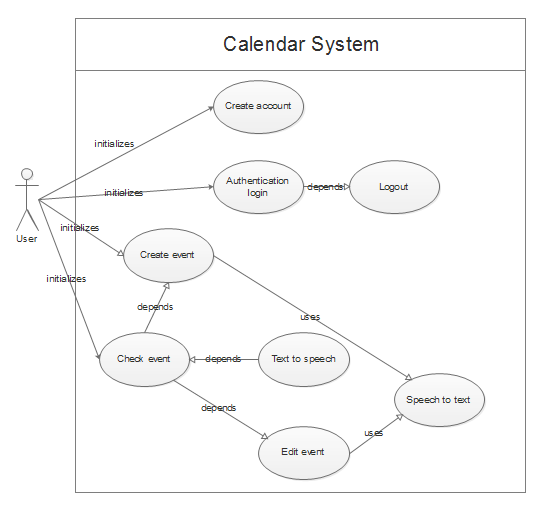


FIgure 2.3.2.3. Use case diagram for client side

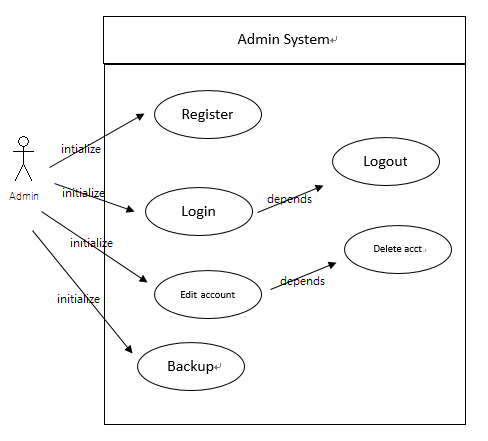


FIgure 2.3.2.3. Use case diagram for admin side

2.3.2.3. Data Flow Diagram

The Data Flow Diagram is to clarify all the function in the database of event and help to explain more about the part of the use case diagram.

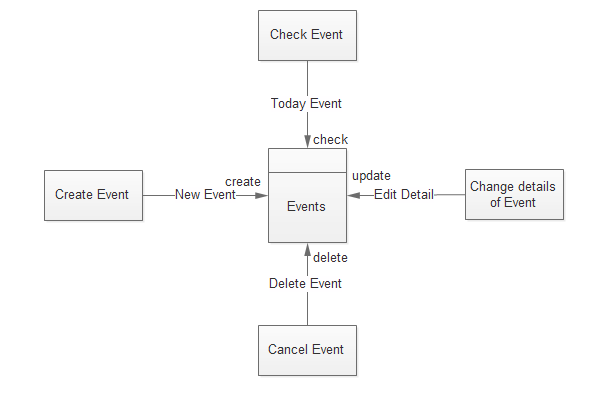


FIgure 2.3.2.4 Data Flow Diagram for Client Side

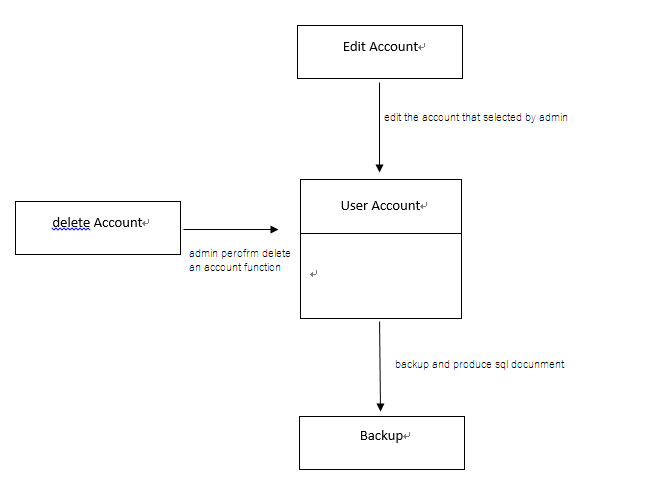


FIgure 2.3.2.4 Data Flow Diagram for Admin Side

Client Side:

|  |  |
| --- | --- |
| Data structure name | New Event |
| Alias |  |
| Composition | Id + Subject + Location + Description + StartTime + EndTime + IsAllDayEvent |

|  |  |
| --- | --- |
| Data structure name | Delete Event |
| Alias |  |
| Composition | Id + Subject + Location + Description + StartTime + EndTime + IsAllDayEvent |

|  |  |
| --- | --- |
| Data structure name | Edit Detail |
| Alias |  |
| Composition | Id + Subject + Location + Description + StartTime + EndTime + IsAllDayEvent |

|  |  |
| --- | --- |
| Data structure name | Today Event |
| Alias |  |
| Composition | Id + Subject + Location + Description + StartTime + EndTime + IsAllDayEvent |

Admin Side:

|  |  |
| --- | --- |
| Data structure name | Edit Account |
| Alias |  |
| Composition | Name+Email |

|  |  |
| --- | --- |
| Data structure name | Delete Event |
| Alias |  |
| Composition | Name+Email |

|  |  |
| --- | --- |
| Data structure name | Backup |
| Alias |  |
| Composition | Name+Email+Password+Id + Subject + Location + Description + StartTime + EndTime + IsAllDayEvent |

|  |  |
| --- | --- |
| Data structure name | Restore |
| Alias |  |
| Composition | Name+Email+Password+Id + Subject + Location + Description + StartTime + EndTime + IsAllDayEvent |

2.3.3.Use Case Narrative for different activity

These are narrative process described the detailed process for system running and the process of how system and Client ‘s responses from different cases,as well as the triggering processes and other important elements related to the Use case model.

2.3.3.1.Create Account Activity

|  |  |  |  |
| --- | --- | --- | --- |
| Use-Case Name: | Create account | | Use case type  Business Requirement |
| Use-Case ID: | FYP-001 | |
| Priority: | High | |
| Source: |  | |
| Primary System Actor: | Client | | |
| Primary Business Actor: | Client | | |
| Other participating Actors: |  | | |
| Other Interested Stakeholders: |  | | |
| Description: | This use case describes the event of a new client register to the system | | |
| Precondition: | Nope | | |
| Trigger: | The use case is initiated when the user selects this option from the user interface | | |
| Typical Course of Events: | Actor Action | System Response | |
| Step 1: This use case is initiated when the user’s click    Step 3: The user input he’s email to account field    Step 5: The user inputs he’s password    Step 7: The user inputs he’s password in password confirm field    Step 9: The user clicks Register button | Step 2: The system responds by displaying register page    Step 4: The system check the account input is valid (an email) or not and responds by displaying a tick    Step 6: The system check the password input is valid (long than 8 length) or not and responds by displaying a tick and password confirm field can be inputted    Step 8: The system check the password and password confirm is same or not, if same responds by displaying a tick    Step 10: The system stores those data into database and responds by displaying login page | |
| Alternate Courses: | Alt Step 4a, 6a, 8a: If the user input invalidly, the system will responds with a cross.  Alt Step 3a, 5a, 7a: If the user do not input all the field, the system will responds by displaying a pop-up message to hints user. | | |
| Conclusion: | This use case concludes when the system saved the registration information | | |
| Postcondition: | Nope | | |
| Business Rules: | Nope | | |
| Implementation Constraints and Specifications: | Web programming and Database to be used so clients can use it easily, and client’s input can be stored | | |
| Assumptions: | Nope | | |
| Open Issues: | Nope | | |

2.3.3.2.Enter account

|  |  |  |  |
| --- | --- | --- | --- |
| Use-Case Name: | Enter account | | Use case type  Business Requirement |
| Use-Case ID: | FYP-002 | |
| Priority: | High | |
| Source: |  | |
| Primary System Actor: | Client | | |
| Primary Business Actor: | Client | | |
| Other participating Actors: |  | | |
| Other Interested Stakeholders: |  | | |
| Description: | This use case describes the event of a client log-on to the system with their account | | |
| Precondition: | The user must have their own account | | |
| Trigger: | The use case is initiated when the user wants to use the system | | |
| Typical Course of Events: | Actor Action | System Response | |
| Step 1: This use case is initiated when user go to the web page    Step 3: The user input he’s account and password | Step 2: The system responds by displaying the login page    Step 4: The system check the inputted account and password check whether the account is existed or not and responds by displaying index page | |
| Alternate Courses: | Alt Step 4a: If the input is invalid, the system will responds by displaying error message. | | |
| Conclusion: | This use case concludes when the user login success | | |
| Postcondition: | Nope | | |
| Business Rules: | Nope | | |
| Implementation Constraints and Specifications: | Web programming and Database to be used so clients can use it easily, and client’s input can be stored | | |
| Assumptions: | User registered an account | | |
| Open Issues: | Nope | | |

2.3.3.3.Check Event

|  |  |  |  |
| --- | --- | --- | --- |
| Use-Case Name: | Check event | | Use case type  Business Requirement |
| Use-Case ID: | FYP-003 | |
| Priority: | High | |
| Source: |  | |
| Primary System Actor: | Client | | |
| Primary Business Actor: | Client | | |
| Other participating Actors: |  | | |
| Other Interested Stakeholders: |  | | |
| Description: | This use case describes the event of a client view events in their calendar | | |
| Precondition: | The user must have previously logged on so that system can identify the user as a particular client | | |
| Trigger: | The use case is initiated when the user login to the system | | |
| Typical Course of Events: | Actor Action | System Response | |
| Step 1: This use case is initiated when the user login | Step 2: The system responds by displaying calendar page and event | |
| Alternate Courses: | Step 3: The user can choose display method (by day or by month) | | |
| Conclusion: | This use case concludes when the system loaded and displayed event information | | |
| Postcondition: | Nope | | |
| Business Rules: | Nope | | |
| Implementation Constraints and Specifications: | Web programming and Database to be used so clients can use it easily, and client’s event can be displayed | | |
| Assumptions: | User registered an account | | |
| Open Issues: | Nope | | |

2.3.3.4.Text to Speech

|  |  |  |  |
| --- | --- | --- | --- |
| Use-Case Name: | Text to speech | | Use case type  Business Requirement |
| Use-Case ID: | FYP-004 | |
| Priority: | High | |
| Source: |  | |
| Primary System Actor: | Client | | |
| Primary Business Actor: | Client | | |
| Other participating Actors: |  | | |
| Other Interested Stakeholders: |  | | |
| Description: | This use case describes the event of system play voice to client | | |
| Precondition: | The user must have previously logged on so that system can identify the user as a particular client and user must open the check events interface | | |
| Trigger: | The use case is initiated when the user login | | |
| Typical Course of Events: | Actor Action | System Response | |
| Step 1: This use case is initiated when use login | Step 2: The system get event information from database and transfer it to voice and responds by playing the voice to user | |
| Alternate Courses: |  | | |
| Conclusion: | This use case concludes when the system play the voice document | | |
| Postcondition: | Nope | | |
| Business Rules: | Nope | | |
| Implementation Constraints and Specifications: | Web programming and Database to be used so clients can use it easily, and client’s event information can be accessed | | |
| Assumptions: | User registered an account, User’s device has speaker | | |
| Open Issues: | Nope | | |

2.3.3.5.Enter Event

|  |  |  |  |
| --- | --- | --- | --- |
| Use-Case Name: | Enter event | | Use case type  Business Requirement |
| Use-Case ID: | FYP-005 | |
| Priority: | High | |
| Source: |  | |
| Primary System Actor: | Client | | |
| Primary Business Actor: | Client | | |
| Other participating Actors: |  | | |
| Other Interested Stakeholders: |  | | |
| Description: | This use case describes the event of a client add new event into their calendar | | |
| Precondition: | The user must have previously logged on so that system can identify the user as a particular client and user must open the check events interface | | |
| Trigger: | The use case is initiated when the user selects this option from the user interface | | |
| Typical Course of Events: | Actor Action | System Response | |
| Step 1: This use case is initiated when the user click        Step 3: The user may choose the edit detail option        Step 5: The user inputs all the event information and click the save button. | Step 2: The system responds by displaying a dialog box with title input, save and edit detail.    Step 4: The system responds by displaying a pop-up window with event detail, save and cancel button.    Step 6: The system saves those information and pop up a success message | |
| Alternate Courses: | Alt Step 3a: If the user click the save button, the system will saves the title and other ‘detail’ information that by system default. | | |
| Conclusion: | This use case concludes when the system saved the event information | | |
| Postcondition: | Nope | | |
| Business Rules: | Nope | | |
| Implementation Constraints and Specifications: | Web programming and Database to be used so clients can use it easily, and client’s input can be stored | | |
| Assumptions: | User registered an account | | |
| Open Issues: | Nope | | |

2.3.3.6.Edit Event

|  |  |  |  |
| --- | --- | --- | --- |
| Use-Case Name: | Edit event | | Use case type  Business Requirement |
| Use-Case ID: | FYP-006 | |
| Priority: | High | |
| Source: |  | |
| Primary System Actor: | Client | | |
| Primary Business Actor: | Client | | |
| Other participating Actors: |  | | |
| Other Interested Stakeholders: |  | | |
| Description: | This use case describes the event of a client edit existing event in their calendar | | |
| Precondition: | The user must have previously logged on so that system can identify the user as a particular client, user must open check events interface | | |
| Trigger: | The use case is initiated when the user selects this option from the user interface | | |
| Typical Course of Events: | Actor Action | System Response | |
| Step 1: This use case is initiated when this user click      Step 3: The user may edit the detail and click Save button | Step 2: The system responds by displaying pop-up window with event detail    Step 4: The system save the details and responds by displaying a success message | |
| Alternate Courses: | Alt Step 3a: If the user click the close button, the system do nothing.  Alt Step 3b: If the user click the delete button, the system will display a confirm message, if user click OK button, the event will be deleted. | | |
| Conclusion: | This use case concludes when the system saved the event information | | |
| Postcondition: | Nope | | |
| Business Rules: | Nope | | |
| Implementation Constraints and Specifications: | Web programming and Database to be used so clients can use it easily, and client’s input can be stored | | |
| Assumptions: | User registered an account, User has inputted event | | |
| Open Issues: | Nope | | |

2.3.3.7.Speech to text

|  |  |  |  |
| --- | --- | --- | --- |
| Use-Case Name: | Speech to text | | Use case type  Business Requirement |
| Use-Case ID: | FYP-007 | |
| Priority: | High | |
| Source: |  | |
| Primary System Actor: | Client | | |
| Primary Business Actor: | Client | | |
| Other participating Actors: |  | | |
| Other Interested Stakeholders: |  | | |
| Description: | This use case describes the event of a client speak out words to input data | | |
| Precondition: | The user must have previously logged on so that system can identify the user as a particular client | | |
| Trigger: | The use case is initiated when the user login | | |
| Typical Course of Events: | Actor Action | System Response | |
| Step 1: This use case is initiated when user login    Step 2: User can speak command to interact with the system | Step 3: The system transfer those voice input the text and responds by executing requested action or input | |
| Alternate Courses: |  | | |
| Conclusion: | This use case concludes when the system responds those commands | | |
| Postcondition: | Nope | | |
| Business Rules: | Nope | | |
| Implementation Constraints and Specifications: | Web programming and Database to be used so clients can use it easily, and client’s input can be stored | | |
| Assumptions: | User registered an account and User has voice input device connected | | |
| Open Issues: | Nope | | |

2.3.3.8.Logout

|  |  |  |  |
| --- | --- | --- | --- |
| Use-Case Name: | Logout | | Use case type  Business Requirement |
| Use-Case ID: | FYP-008 | |
| Priority: | High | |
| Source: |  | |
| Primary System Actor: | Client | | |
| Primary Business Actor: | Client | | |
| Other participating Actors: |  | | |
| Other Interested Stakeholders: |  | | |
| Description: | This use case describes the event of a client to log off their account | | |
| Precondition: | The user must have previously logged on so that system can identify the user as a particular client | | |
| Trigger: | The use case is initiated when the user selects this option from the user interface | | |
| Typical Course of Events: | Actor Action | System Response | |
| Step 1: This use case is initiated when this user click | Step 2: The system responds by displaying a login interface | |
| Alternate Courses: |  | | |
| Conclusion: | This use case concludes when the system display login page | | |
| Postcondition: | Nope | | |
| Business Rules: | Nope | | |
| Implementation Constraints and Specifications: | Web programming to be used so clients can use it easily | | |
| Assumptions: | User registered an account | | |
| Open Issues: | Nope | | |

2.4 Data Store

Data stores acts as a temporary repository of data, their role aims ar reading the form and scripting in store for processing data flow. Following table concluded the required data flows to process the function required under subsystems.

|  |  |
| --- | --- |
| Data Store ID | D1 |
| Data Store Name | Create Activity |
| Alias |  |
| Composition | {Subject+Subject\_Color+Start\_Time+End\_Time+Location+Event\_Detail+Voice\_Recognition+save+cancel} |
| Organization |  |

|  |  |
| --- | --- |
| Data Store ID | D2 |
| Data Store Name | Voice Output Function |
| Alias |  |
| Composition | {Subject+Start\_Time+End\_Time+Location+Event\_Detail} |
| Organization |  |

|  |  |
| --- | --- |
| Data Store ID | D3 |
| Data Store Name | Cancel Data or event |
| Alias |  |
| Composition | {Subject+Start\_Time+End\_Time+Location+Event\_Detail+cancel} |
| Organization |  |

|  |  |
| --- | --- |
| Data Store ID | D4 |
| Data Store Name | Login Function |
| Alias |  |
| Composition | {login\_id+login\_password} |
| Organization |  |

|  |  |
| --- | --- |
| Data Store ID | D5 |
| Data Store Name | Register Function |
| Alias |  |
| Composition | {login\_id+login\_name+login\_password+login\_password\_confirm+confirm} |
| Organization |  |

2.5 Process Description

Process Description The following tables specify the logical description for each of the function primitive process in location based system.

|  |  |
| --- | --- |
| Process No | 1.1 |
| Process Name | Create Activity |
| Input | scripted data |
| Output | scripted result |
| Description | GET scripted\_Data FROM User Input IN mobile/desktop/portable\_dev SEND scripted result TO view |

|  |  |
| --- | --- |
| Process No | 1.2 |
| Process Name | Create Activity |
| Input | scripted data |
| Output | scripted result |
| Description | GET scripted\_Data FROM User Input IN mobile/desktop/portable\_dev SEND scripted result TO view |

|  |  |
| --- | --- |
| Process No | 1.3 |
| Process Name | Delete Activity |
| Input | User\_Input |
| Output | scripted\_result |
| Description | GET User Input FROM User Input IN mobile/desktop/portable\_dev DELETE scripted\_result FROM mobile/desktop/portable\_dev |

|  |  |
| --- | --- |
| Process No | 1.4 |
| Process Name | Voice Output |
| Input | scripted\_result |
| Output | Voice\_output |
| Description | GET scripted\_result FROM User Input IN mobile/desktop/portable\_dev SEND voice\_output to view |

|  |  |
| --- | --- |
| Process No | 1.5 |
| Process Name | Voice Output |
| Input | scripted\_result |
| Output | Voice\_output |
| Description | GET scripted\_result FROM User Input IN mobile/desktop/portable\_dev SEND voice\_output to view |

|  |  |
| --- | --- |
| Process No | 1.6 |
| Process Name | Voice Input |
| Input | Voice\_input\_data |
| Output | Voice\_output |
| Description | GET voice\_input\_data FROM User Voice IN mobile/desktop/portable\_dev SEND voice\_input\_data to scripted data |

|  |  |
| --- | --- |
| Process No | 1.7 |
| Process Name | Voice Control |
| Input | Voice\_input\_data |
| Output | action |
| Description | GET voice\_input\_data FROM User Voice IN mobile/desktop/portable\_dev SEND voice\_input\_data to action |

|  |  |
| --- | --- |
| Process No | 1.8 |
| Process Name | Voice Control |
| Input | Voice\_input\_data |
| Output | action |
| Description | GET voice\_input\_data FROM User Voice IN mobile/desktop/portable\_dev SEND voice\_input\_data to action |

|  |  |
| --- | --- |
| Process No | 1.9 |
| Process Name | Now time indication |
| Input | / |
| Output | redline indicated on the time slot |
| Description | / |

|  |  |
| --- | --- |
| Process No | 1.10 |
| Process Name | Day view |
| Input | Input\_day |
| Output | day\_view |
| Description | GET Input\_Day FROM User IN mobile/desktop/portable\_dev SEND dayview to view |

|  |  |
| --- | --- |
| Process No | 1.11 |
| Process Name | Month view |
| Input | Input\_Month |
| Output | Month\_view |
| Description | GET Input\_Month FROM User IN mobile/desktop/portable\_dev SEND Monthview to view |

|  |  |
| --- | --- |
| Process No | 2.01 |
| Process Name | Login |
| Input | user\_name+user\_pw |
| Output | Login Result |
| Description | GET user\_name+user\_pw FROM User IN mobile/desktop/portable\_dev SEND Login Result to view |

|  |  |
| --- | --- |
| Process No | 2.02 |
| Process Name | Logout |
| Input | logout |
| Output | Login\_Page |
| Description | GET logout FROM User IN mobile/desktop/portable\_dev SEND Login\_Page to view |

|  |  |
| --- | --- |
| Process No | 3.01 |
| Process Name | Registration |
| Input | user\_name+user\_pw+user\_confirm\_result |
| Output | Register Result |
| Description | GET user\_name+user\_pw+user\_confirm\_result FROM User IN mobile/desktop/portable\_dev SEND Register Result to view |

2.6 External Entities Description

|  |  |
| --- | --- |
| External Entity ID | E1 |
| External Entity Name | User |
| Alias | Visual patient,people worked busily |
| Description | User whose want to login in through system from verification of system. |

|  |  |
| --- | --- |
| External Entity ID | E2 |
| External Entity Name | Administrator |
| Alias |  |
| Description | Managing the datum among difference of users |

|  |  |
| --- | --- |
| External Entity ID | E3 |
| External Entity Name | Google Server(Voice Input) |
| Alias |  |
| Description | A server maintained by google that could converted the voice input from users to lexical word. |

|  |  |
| --- | --- |
| External Entity ID | E4 |
| External Entity Name | Google Server(Voice Output) |
| Alias |  |
| Description | A server maintained by google that could converted the lexical word from database to Voice Output |

|  |  |
| --- | --- |
| External Entity ID | E5 |
| External Entity Name | Google Server(Voice Control) |
| Alias |  |
| Description | A server maintained by google that could converted the voice output to an array of consequential actions |

3.DATABASE DESIGN

In this chapter, there are four sections to describe the database design, such as database design, data decomposition, entities description and data description.

In section 3.1, the structure of the database will be shown.

In section 3.2, the table diagram will be listed will attributes’ description.

In section 3.3, entity description will be provided with table will be listed in details.

In section 3.4, there are overall attribute to show.

3.1 Data Analysis

The data model for Voice Calendar is shown as the following.

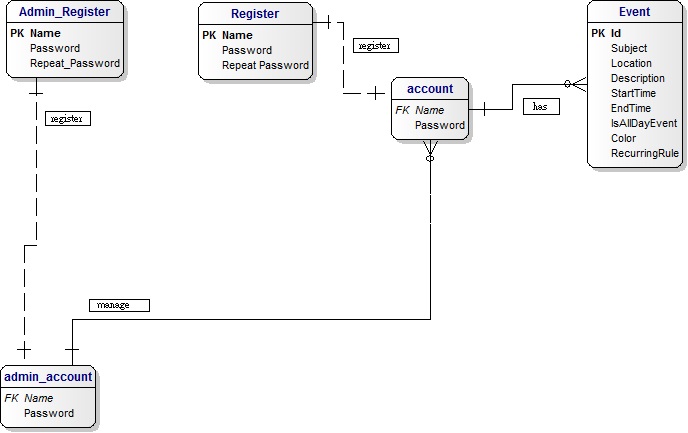


Figure 3.1 The EAR Diagram of the Voice Calendar

3.2 Table Definition and Description

|  |  |  |
| --- | --- | --- |
| Name | Number of columns | Description |
| Register | 3 | For storing new users’ name and password |
| Account | 2 | For storing users’ name and password |
| Event | 9 | For storing the details of an event |

Figure 3.2 Table of Data Diagram of Voice Calendar

3.3 Data Description

3.3.1 Register

|  |  |  |  |
| --- | --- | --- | --- |
| Column name | Data type | Not NULL | Remark |
| Name | VARCHAR(40) | Yes | PK |
| Password | VARCHAR(40) | Yes |  |
| RepeatPassword | VARCHAR(40) | Yes |  |

Figure 3.3 Attributes of the entity - Register

3.3.2 Account

|  |  |  |  |
| --- | --- | --- | --- |
| Column name | Data type | Not NULL | Remark |
| Name | VARCHAR(40) | Yes | PK, FK |
| Password | VARCHAR(40) | Yes |  |

Figure 3.4 Attributes of the entity - Account

3.3.3 Event

|  |  |  |  |
| --- | --- | --- | --- |
| Column name | Data type | Not NULL | Remark |
| Id | INTEGER(11) | Yes | PK |
| Subject | VARCHAR(1000) | No |  |
| Location | VARCHAR(200) | No |  |
| Description | VARCHAR(255) | No |  |
| StartTime | DATETIME | No |  |
| EndTime | DATETIME | No |  |
| IsAllDayEvent | SMALLINT(6) | Yes |  |
| Color | VARCHAR(200) | No |  |
| RecurringRule | VARCHAR(500) | No |  |

Figure 3.5 Attributes of the entity - Event

3.3.4. Administrators Register

|  |  |  |  |
| --- | --- | --- | --- |
| Column name | Data type | Not NULL | Remark |
| AdmName | VARCHAR(40) | Yes | PK |
| Password | VARCHAR(40) | Yes |  |
| RepeatPassword | VARCHAR(40) | Yes |  |

Figure 3.6 Attributes of the entity - Admin Register

3.3.5. Administrators Account

|  |  |  |  |
| --- | --- | --- | --- |
| Column name | Data type | Not NULL | Remark |
| AdmName | VARCHAR(40) | Yes | PK, FK |
| Password | VARCHAR(40) | Yes |  |

Figure 3.7 Attributes of the entity - Admin Account

3.4 Overall attributes of Voice Calendar

Client Side

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column name | Table | Key | Data type | Not NULL |
| Color | Event |  | VARCHAR(200) | No |
| Description | Event |  | VARCHAR(255) | No |
| EndTime | Event |  | DATETIME | No |
| Id | Event | PK | INTEGER(11) | Yes |
| IsAllDayEvent | Event |  | SMALLINT(6) | Yes |
| Location | Event |  | VARCHAR(200) | No |
| Name | Register | PK | VARCHAR(40) | Yes |
| Name | Account | PK, FK | VARCHAR(40) | Yes |
| Password | Register |  | VARCHAR(40) | Yes |
| Password | Account |  | VARCHAR(40) | Yes |
| RepeatPassword | Register |  | VARCHAR(40) | Yes |
| RecurringRule | Event |  | VARCHAR(500) | No |
| StartTime | Event |  | DATETIME | No |
| Subject | Event |  | VARCHAR(1000) | No |

Figure 3.8 Client Side Database

Admin Side

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| AdmName | Admin Register |  | VARCHAR(40) | Yes |
| AdmName | Admin Account |  | VARCHAR(40) | Yes |
| Password | Admin Register |  | VARCHAR(40) | Yes |
| Password | Admin Account |  | VARCHAR(40) | Yes |
| RepeatPassword | Admin Register |  | VARCHAR(40) | Yes |

Figure 3.9 Admin Side Database

Keys:

PK: Primary Key

FK: Foreign Key

4. System Design

These pages are shows the elements of each pages and their actual functions clearly by chart illustrated.

4.1 Calendar

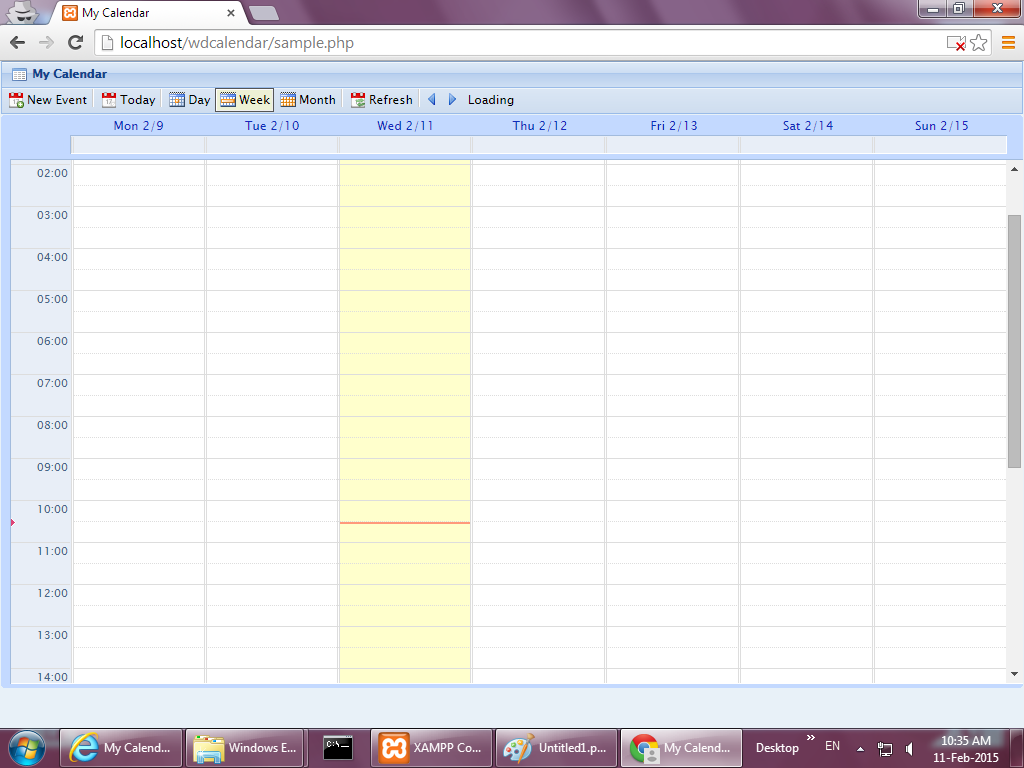


Figure 4.1 The page of the calendar

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | New Event | Button | User can create event |
| 2 | Today | Button | User can show today time slot |
| 3 | Day | Button | User can show the chosen day time slot |
| 4 | Week | Button | User can show the weekly time slot |
| 5 | Month | Button | User can show the monthly timetable |
| 6 | Refresh | Button | User can update the information of the calendar |
| 7 | Time Slot | Box | User can create event by clicking exactly time slot |

4.2 Create event

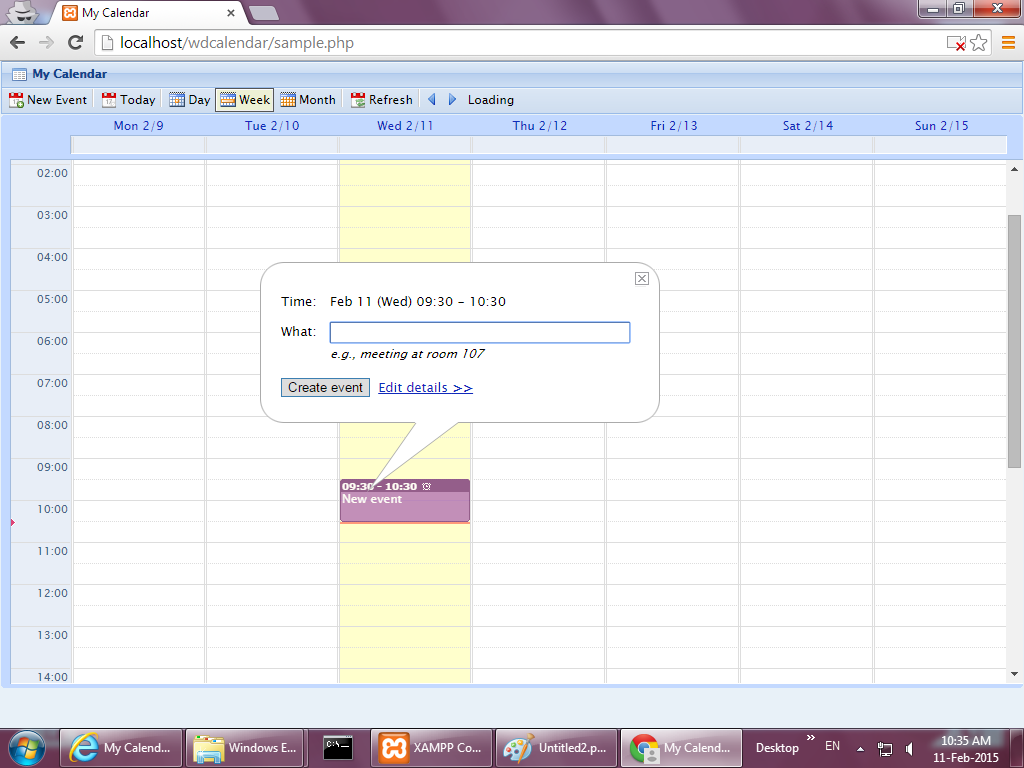


Figure 4.2 Create an event by clicking time slot

Notation: After clicking the exactly time slot from the calendar, users will be directed to this page to create the title of the event.

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | What | Text Box | User can create the title of an event |

4.3 Create and edit details of event

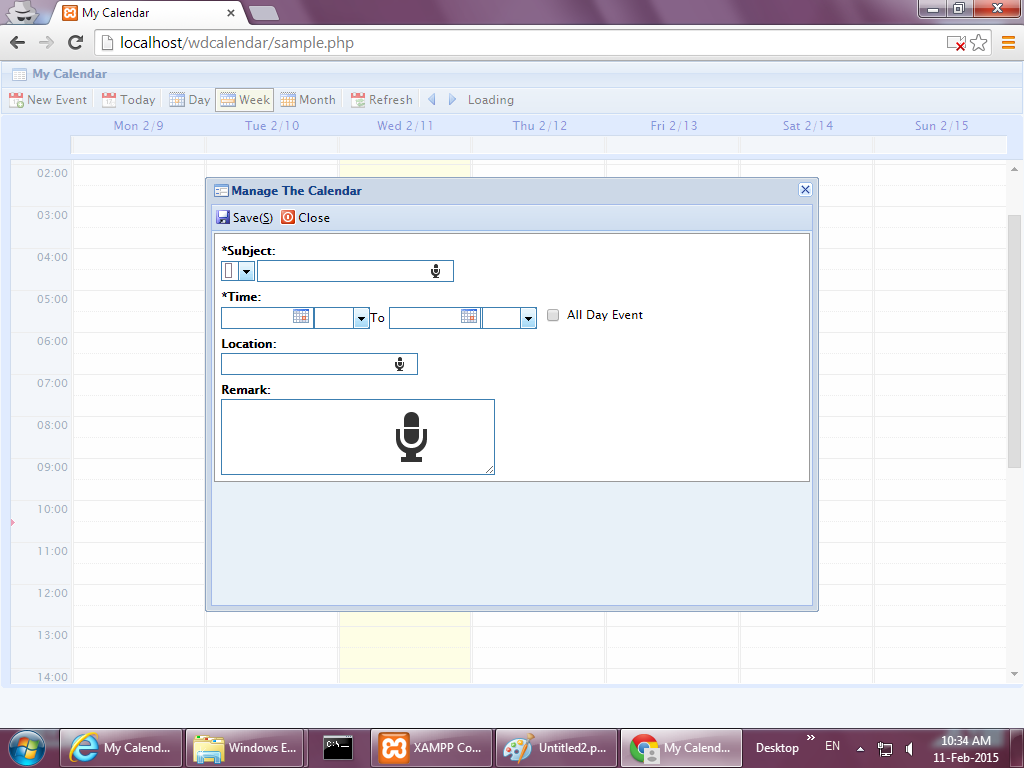


Figure 4.3 Editing details of an event

Notation: After clicking edit details, users will be directed to this page to edit details of events.

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | Colour | Spinner | User can choose the colour of the title of an event |
| 2 | Title | Text Box | User can click the mic button to create the title through the voice input |
| 3 | Starting time | Spinner | User can choose the starting time and date |
| 4 | Ending time | Spinner | User can choose the ending time and date |
| 5 | All-Day | Check Box | User can click this check box to set an event be the all-day event |
| 6 | Location | Text Box | User can click the mic button to create the location of an event through the voice input |
| 7 | Remark | Text Box | User can click the mic button to create the special requirement of an event through the voice input |

4.4 Day schedule

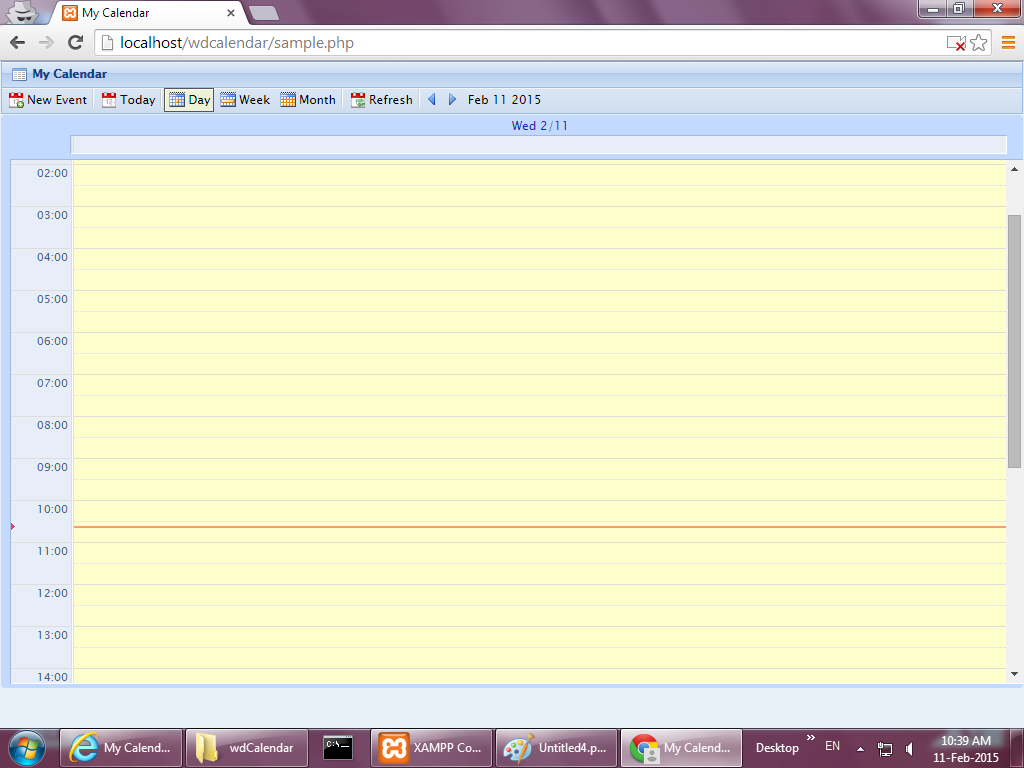


Figure 4.4 The day schedule

Notation: After clicking the day button, users will be directed to this page to show out events on that day

4.5 Month schedule

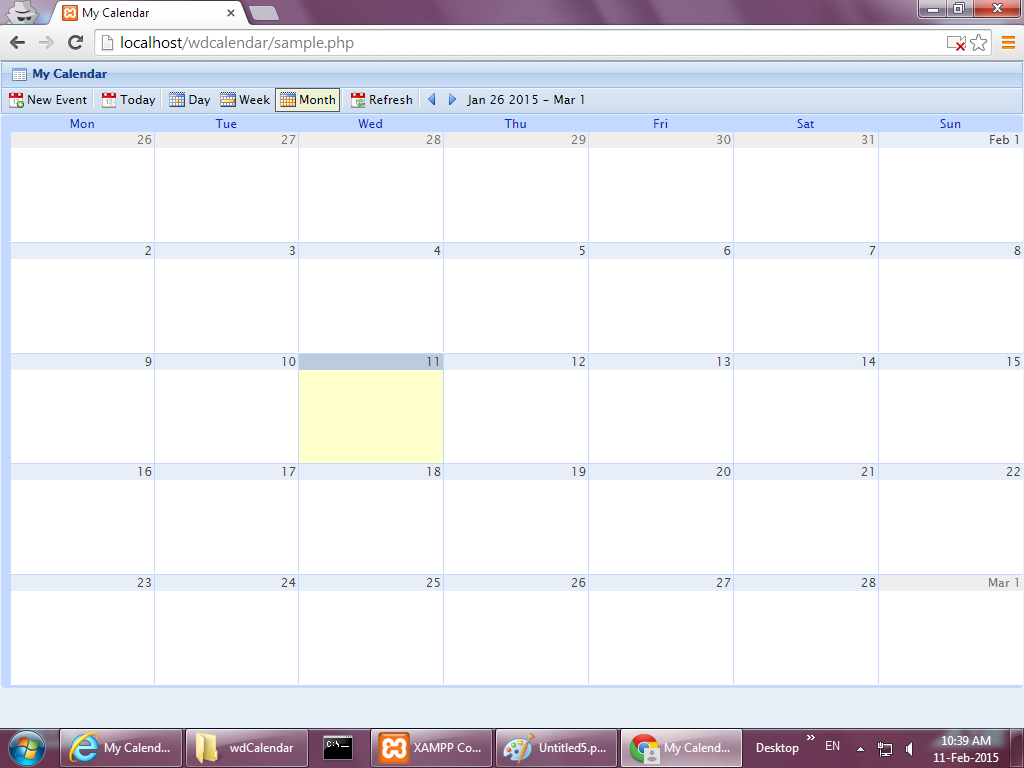


Figure 4.5 The calendar that shows a month

Notation: After clicking the month button, users will be directed to this page to see what events they have of a month

5 TEST PLAN

5.1 Objective

The objective of this test plan is evaluate the application quality under different test methods. Those test can also check the accuracy of achieve application objective and requirement. This test plan will include testing detail, testing manner and testing schedule.

Reference Documents

The following are supplementary documents for understand the test if have problem:

1. Project Plan

2. System Specification

3. User manuals

5.2 Testing Plan

The following listed item will be tested under this plain:

1. Voice Calendar

2. Administrator system

5.3 Testing Scope

1. Operation Test

The Operation test will focus on the performance of the test products under

normal condition.

2. System Test

The System test will verify the test products are met the project requirements

and objectives.

3. Database Test

The Database test will test the operation of database and check whether the

database can run the queries properly or not.

5.4 Testing Strategy

There are many different testing exists for testing. We choose the following two testing strategies for our application testing.

1. Module-based Testing

Module-based testing is an abstract testing. This test evaluates the products without any knowledge of the internal mechanism of the system or component. The test separate a complex system to many smaller models for testing and test those methods individually. It can test every function and find out the bugs in each part of the system.

2. Case-based testing

Case-based testing is a test based on the case defined by tester. It can simulate the real world situation of using the application, so that proper error can be prevented by the testing.

5.5 Module-based Testing

1. Voice Calendar

i. Login process

|  |  |
| --- | --- |
| Action A | Press Correct Account and Password. Then press Login button. |
| Expect result | Show the corresponding account’s calendar page. |

|  |  |
| --- | --- |
| Action B | Press Login button without input either account, password or both. |
| Expect result | Stay at login page and show error message. |

|  |  |
| --- | --- |
| Action C | Press Login button with wrong account or password. |
| Expect result | Stay at login page and show error message. |

ii. Registration process

|  |  |
| --- | --- |
| Action A | Input username, email, password and repeat password correctly, then press Register button |
| Expect result | A new account created |

|  |  |
| --- | --- |
| Action B | Missing at least one item in username, email, password and repeat password, then press Register button |
| Expect result | Remain in registration page and show error message |

|  |  |
| --- | --- |
| Action C | Input username with symbol or space OR  Input username with only one character OR  Input username with 65 character,  then press Register button |
| Expect result | Remain in registration page and show error message |

|  |  |
| --- | --- |
| Action D | Inputted password and repeat password are different |
| Expect result | Remain in registration page and show error message |

iii. Create event

|  |  |
| --- | --- |
| Action A | Click the calendar and click ‘Edit details’ or Click the ‘New event’ button, and input event subject, start and end time, location, remarks. Press the ‘save’ button |
| Expect result | Create event successfully and event displayed on calendar |

|  |  |
| --- | --- |
| Action B | Missing or not select start and end time during Action A |
| Expect result | Remain in that page and required to input time |

|  |  |
| --- | --- |
| Action C | Subject is empty during Action A |
| Expect result | Remain in that page and required to input subject |

|  |  |
| --- | --- |
| Action D | Click the calendar, and input event name. Press the ‘Create event’ button |
| Expect result | Create event successfully and event displayed on calendar |

|  |  |
| --- | --- |
| Action E | Do not input event name during Action D |
| Expect result | Remain in that page and required to input event name |

|  |  |
| --- | --- |
| Action F | Voice out ‘Create’, voice out the input content in order from Subject, Time, Location and Remark |
| Expect result | Create event successfully and event displayed on calendar |

iv. Delete event

|  |  |
| --- | --- |
| Action A | Click the existing event, press the ‘Delete’ button and click ‘OK’ from the confirm notice |
| Expect result | Event deleted and show message ‘success’ |

v. Specifying the events

|  |  |
| --- | --- |
| Action A | Click the ‘CHECK YOUR SCHEDULE NOW’ button |
| Expect result | Event will be displayed |

|  |  |
| --- | --- |
| Action B | Voice out ‘speak’ in calendar page |
| Expect result | The computer sound out event name, time, location and remarks of today upcoming event |

vi. Voice Output

|  |  |
| --- | --- |
| Action A | Press the key which will create sound effect |
| Expect result | The computer sound out the corresponding sound |

vii. Voice Input

|  |  |
| --- | --- |
| Action A | Voice out instruction words |
| Expect result | Execute the corresponding action correctly |

viii. Update the events

|  |  |
| --- | --- |
| Action A | Click the existing event, click the ‘Edit details’ button, change the event’s details and click ‘Save’ button |
| Expect result | The event’s details will be changed |

|  |  |
| --- | --- |
| Action B | Press and hold the existing event, move the event within the calendar and release |
| Expect result | The event’s time will be changed |

2. Administrator system

i. Login process

ii. Managing the database

iii. Dump

iv. Restore

5.6 Case-based Testing

1. Case 1

User

Use the application without any network access. User should not able to login the application and always stay at main page.

2. Case 2

User

Use the application with network access. User should able to do registration, login if they have account, enter event to calendar by typing or voice, get the event from calendar by showing text and voice.

5.7 Pass or Fail Criteria

The following is the specification that defines the application is passing or

not:

1. No system failure such as display error and non-functioning function (for

Voice Calendar)

2. No system failure such as display error and non-functioning function (for

Administrator system)

3. The information in the database are inputted correctly and being shown at

Voice Calendar correctly

4. Correct error message provided for incorrect input

5.8 Suspension Criteria

If any defects are found which seriously crash the test progress, the tester

may choose to suspend testing. The following criteria will justify test

suspension:

1. Invalid data provided by administrator for the test

2. System source code contains serious error or limit the progress of testing

5.9 Testing Schedule

Alpha test and beta test will be conducted with this test plan. Alpha test will be carried out in parallel with the system development process. Programmers and system designers can help to test the system development of each process during the process of system development. The alpha test can help to find out the advantages and disadvantages or any bugs for each process of the project.

Beta test will be conducted after the development of the project application is completed. The aim of the test is to examine the system with the live situation, including the target users and system administrators. The beta test can help to find that what does the project miss for the need of target users. Target users and system administrators can participate to test the application.

6. LIMITATIONS & FURTHER DEVELOPMENT

6.1 Limitations

There are two limitations for the project.

6.1.1 Only Google is available to use

For the voice calendar, it is only available to use on Google Chrome this web browser. When users use other web browsers to go to this website, they cannot use those functions which are included with voice. Users can only go to this website through Google Chrome to use those voice functions. This leads the trouble that they need to find Google Chrome to use. If the devices do not have Google Chrome, they need to install it for using this voice calendar.

6.1.2 Cannot remember the option of users

Since the voice calendar is not https for the website, there is a limited security for the system cannot remember the choosing options of users. When users want to use microphone to implement the voice input to the voice calendar, Google Chrome will ask users to choose do they accept to use the microphone of the devices to input the information every time. This leads the trouble that users need to click the accept button every time.

6.2 Further Development

6.2.1 Share the timetable of calendar

For further development, users can share their timetable of the calendar with their friends. When they want to choose available time to have gathering or talking about the project, they can use the calendar to see which of the time slot is available between them. They can see both of their timetable and know each other which time is free for them to choose.