National College of Ireland
BSc in Computing
2017/2018

Deividas Sevcenko
X13114654
X13114654@student.ncirl.ie

Multi-calendar

Technical Report

Table of Contents

Executive Summary

Maximum 300 words. The abstract should mention the problem being addressed, describe the technical solution and briefly report the findings of the evaluation.

1 Introduction

The purpose of this document is to set out the requirements for the development of web application Multi-calendar. Multi-calendar will be designed to fill a gap in the market that no other calendar has. It will allow user create multiple calendars for different companies in one place. The intended customers will be anyone who likes to use online calendar. Initially I would try target anyone who doing multiple jobs.

1.1 Background

Calendar

Main purpose of this application to allow user create multiple calendars for different companies. Also add business partners for each company. Register user could check free and busy time of any user without permission but he should be register in same company. By following this calendar user could easer to plan his day.

Meeting

User is able to create meeting with business partner or multiple business partners. If user send request of meeting then business partner have to accept or reject. By accepting request he will automatically edit his calendar with one extra meeting. If user want to send request of meeting for multiple business partners then he is able to choose two type of meeting. First will be all of group members should accept request other way meeting will be cancelled. Second type one of group members should accept request.

Permission

Owner of calendar is able to give permission for any business partner. If user has permission access business partner calendar he is able to see content of meetings.

1.2 Aims

The aims of the project is to develop a web application that allows user to create multi calendars. For example if person is working in Company A, Company B, and

Company C then he has to maintain three different calendars. This application will allow user to create many calendars he want in one place. With multi-calendar user will be able to share his free and busy time, check any other user availability, arrange meeting. For this project I have chosen one of lecture idea. First I did my research to make sure does this idea doesn't exist.

1.3 Technologies

HTML and CSS

For this project structure and design layout I will use HTML and CSS.

PHP

For this project I will use server side scripting language. PHP scripts allow me manipulate data from database POST, DELETE, UPDATE, GET.

JavaScript

To make web application interactive I will use JavaScript, also I will use JQuery and Ajax for user login and register validation.

Bootstrap and JQuery libraries

HTML, CSS, and JS framework.

Cloud9

Cloud9 is free to use and provides an environment where I can code and test the app during development for no cost. I will use Cloud9 development environment it has integrated Apache and MySQL servers, also allow me do my coding at any machine.

PHPMyAdmin

PHPMyAdmin is free to use and provides graphical user interface for working with MySQL database management system. I will use PHPMyAdmin to store and manipulate data in database.

Microsoft Azure

This project will be deployed by using Microsoft azure.

1.4 Structure

1.4.1 Introduction

This section will explain the background and aims of this project while the technologies section will give a brief explanation of the programming language and devices used to build the application. While structure explains the structure of this document.

1.4.2 System

This section describes requirements, design and architecture. Requirements section give brief explanation of functional requirements, non-functional requirements, data requirements, user requirements, environmental requirements and usability requirements. Design architecture section give brief explanation of implementation, GUI, testing, customer testing and evaluation.

1.4.3 Conclusion

This section gives an overall conclusion on this document and the application.

1.4.4 Further Development

Describes the future opportunities for the application in the near future and any upgrades that can be added to make the application beat any competition in the future if any.

1.4.5 References

Bibliography of used recourses to complete technical report.

1.4.6 Appendix

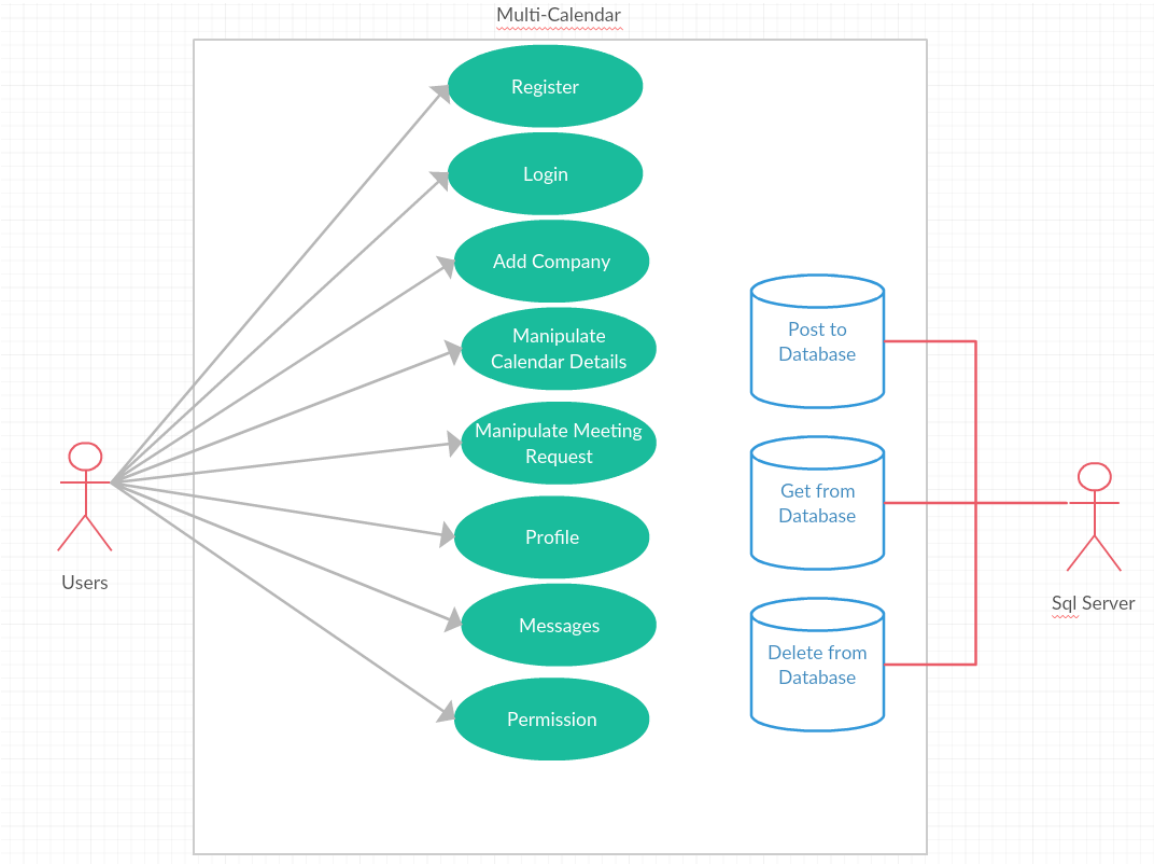
This section contains a copy of the project proposal, the project plan and the monthly journals.

2 System

2.1 Requirements

2.1.1 Functional requirements

2.1.1.1 Use Case Diagram



2.1.1.2 Requirement 1: Register

2.1.1.2.1 Description & Priority

Register is the main priority of the application. It enable to login to application and access all features.

2.1.1.2.2 Use Case

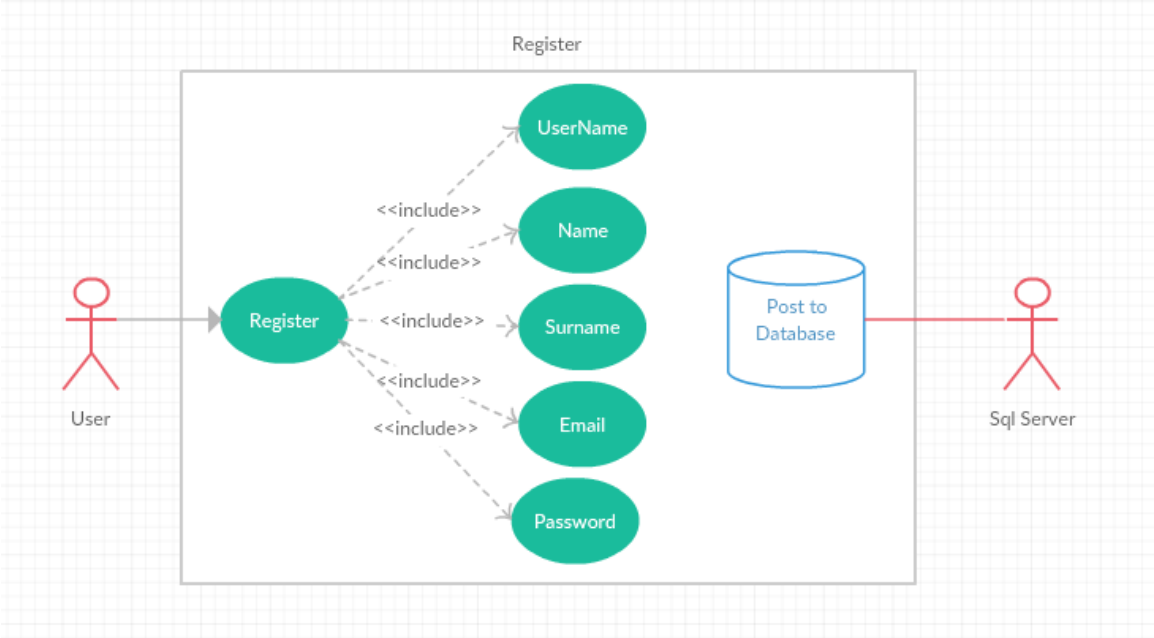
Scope

The scope of this use case is to allow user to register as new user and create account.

Description

This use case describes how to register for new user.

Use Case Diagram



Flow Description

Precondition

User has not yet register.

Activation

This use case starts when the user want to create new account by clicking Register button.

Main flow

1. Use case begins when user press register button
2. User must enter details to register form
3. Submit details
4. System will validate details

Alternate flow

1. Field not completed
   All required fields must be completed.
2. Username or email address already exist
   Username and email must be unique.

Termination

Main flow: User successfully registered.

Alternate flow: User fail to register.

Post condition

Main flow: User reverted to login page.

Alternate flow: User must enter required details.

2.1.1.3 Requirement 2: Login

2.1.1.3.1 Description & Priority

This requirement allow user login into his account and access to all features.

2.1.1.3.2 Use Case

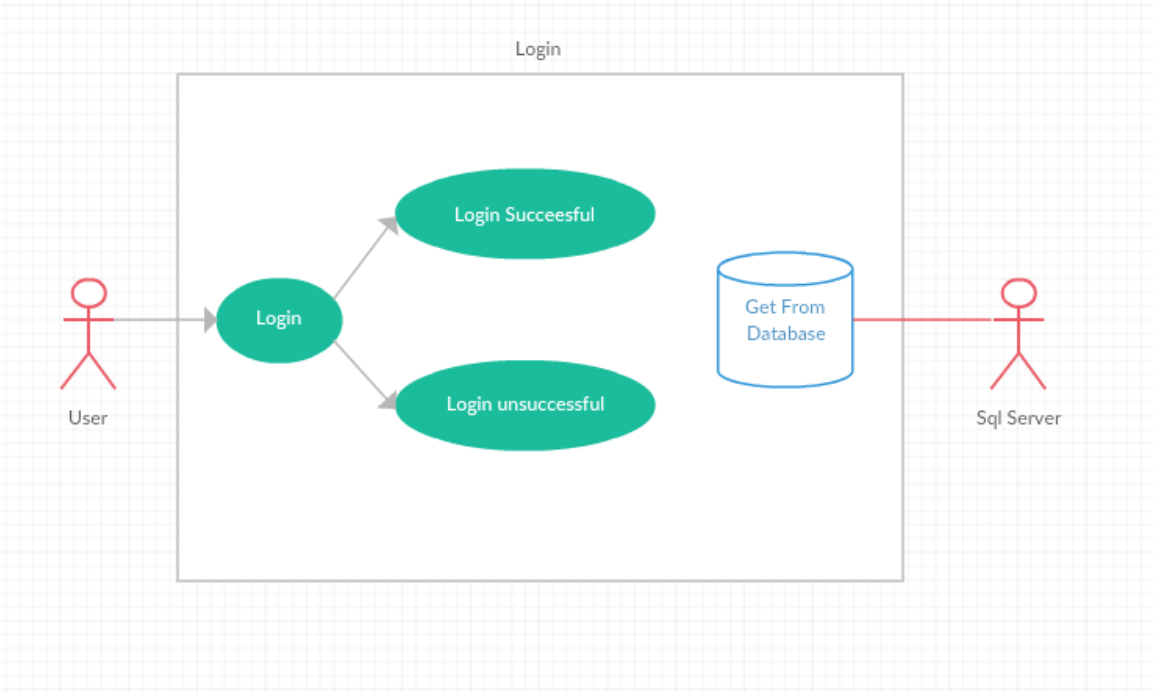
Scope

The scope of this use case is to allow user to access his profile and manipulate multi-calendar.

Description

This use case describes how to login to account for registered user.

Use Case Diagram



Flow Description

Precondition

User must validate his account true email.

Activation

This use case starts when user press login button.

Main flow

1. Use case begins when user press login button
2. User must enter details to login form
3. Submit details
4. System will validate details

Alternate flow

1. Field not completed
   Email and password fields must be completed.
2. Password or email address not exist
   Username and email must be correct.

Termination

Main flow: Login successful.

Alternate flow: Login unsuccessful.

Post condition

Main flow: User can access profile and calendar.

Alternate flow: User reverted to login form.

2.1.1.4 Requirement 3: Add Company

2.1.1.4.1 Description & Priority

This requirement allow user to add new company and allow to see other users in same company.

2.1.1.4.2 Use Case

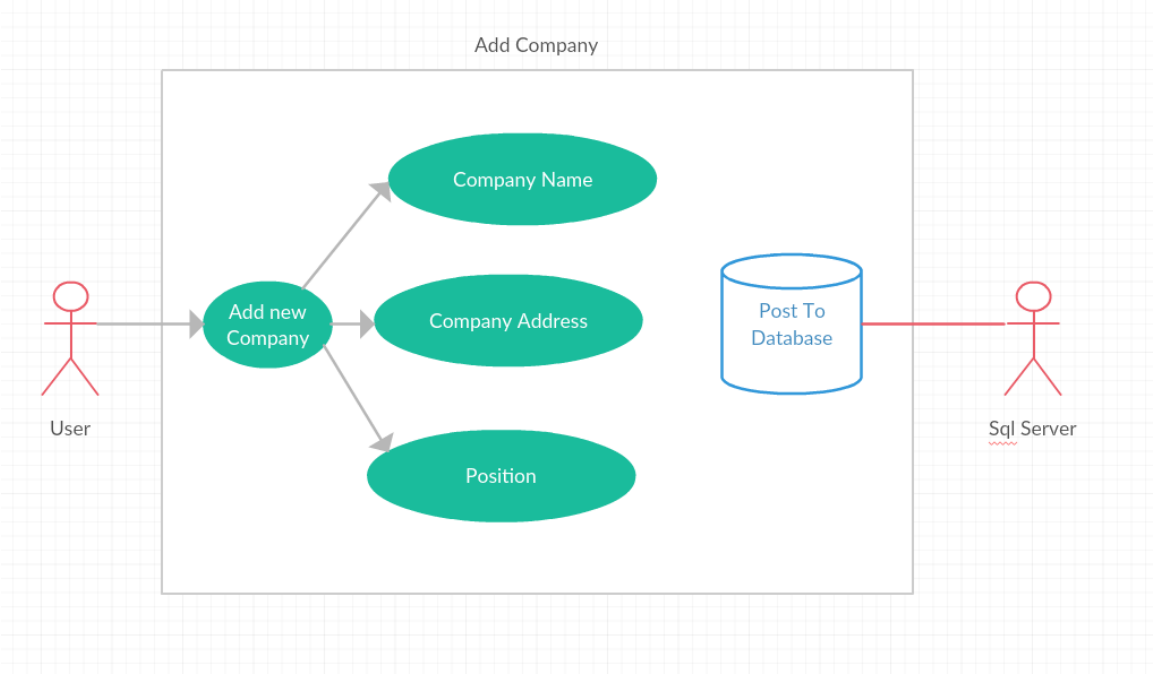
Scope

The scope of this use case is to allow user to add multiple companies

Description

This use case describes the how to add new company.

Use Case Diagram



Flow Description

Precondition

User must be in same company to see other users.

Activation

This use case starts when user in his account press button add new company.

Main flow

1. User case begin when user press add new company button
2. User must enter details of company
3. Submit details
4. System will post data to database

Alternate flow

1. Fields not completed
2. Company should be unique

Termination

Main flow: Company created successful.

Alternate flow: Company was not created.

Post condition

Main flow: User can add meetings for new company.

Alternate flow: User reverted to add new company form.

2.1.1.5 Requirement 4: Manipulate Calendar Details

2.1.1.5.1 Description & Priority

This requirement allow user to manipulate calendar entries. Add new entry to calendar, edit existing entry, and delete unwanted entry.

2.1.1.5.2 Use Case

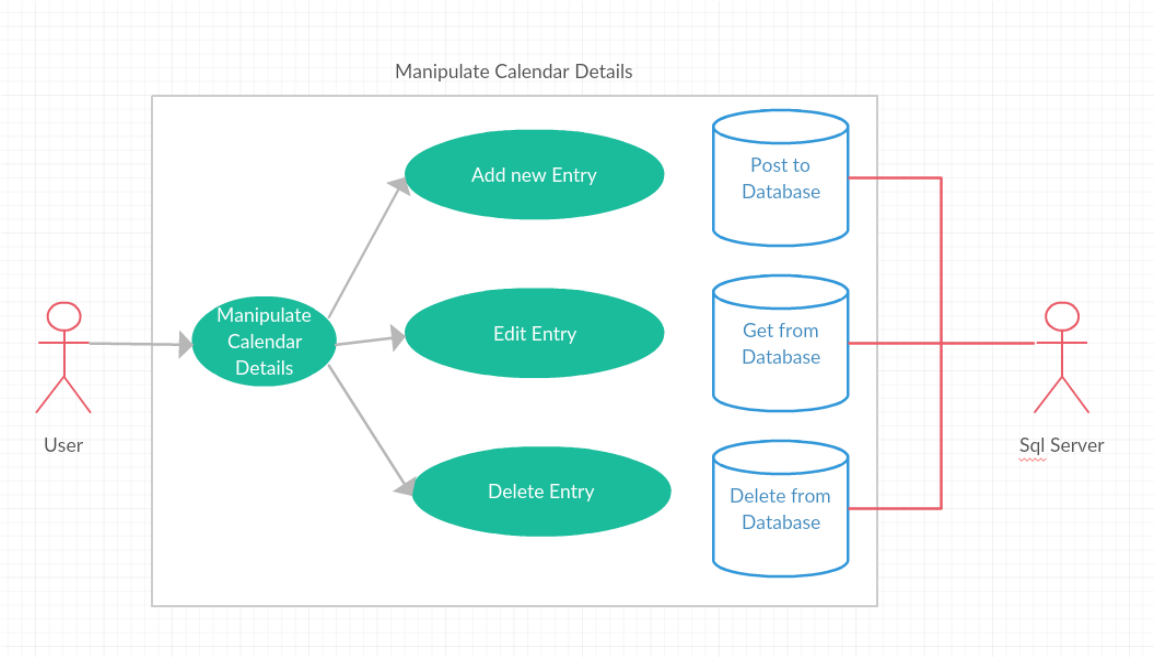
Scope

The scope of this use case is to allow user to manipulate calendar details.

Description

This use case describes how to manipulate details of calendar.

Use Case Diagram



Flow Description

Precondition

User must login to his account to access his calendar.

Activation

This use case starts when user chose one of option by clicking particular date and time.

Main flow

1. User case begin when user press chosen day at calendar
2. User must enter details of entry
3. Submit details
4. System will post data to database

Alternate flow

1. Fields not completed

2. Entry not fit into time frame

Termination

Main flow: Entry was created.

Alternate flow: Entry was not created.

Post condition

Main flow: User could see entry in calendar.

Alternate flow: User reverted to calendar entry form.

2.1.1.6 Requirement 5: Manipulate Meeting Request

2.1.1.6.1 Description & Priority

This requirement allow user manipulate meeting requests. User are able to send meeting request to other user or edit, accept, delete existing meeting request.

2.1.1.6.2 Use Case

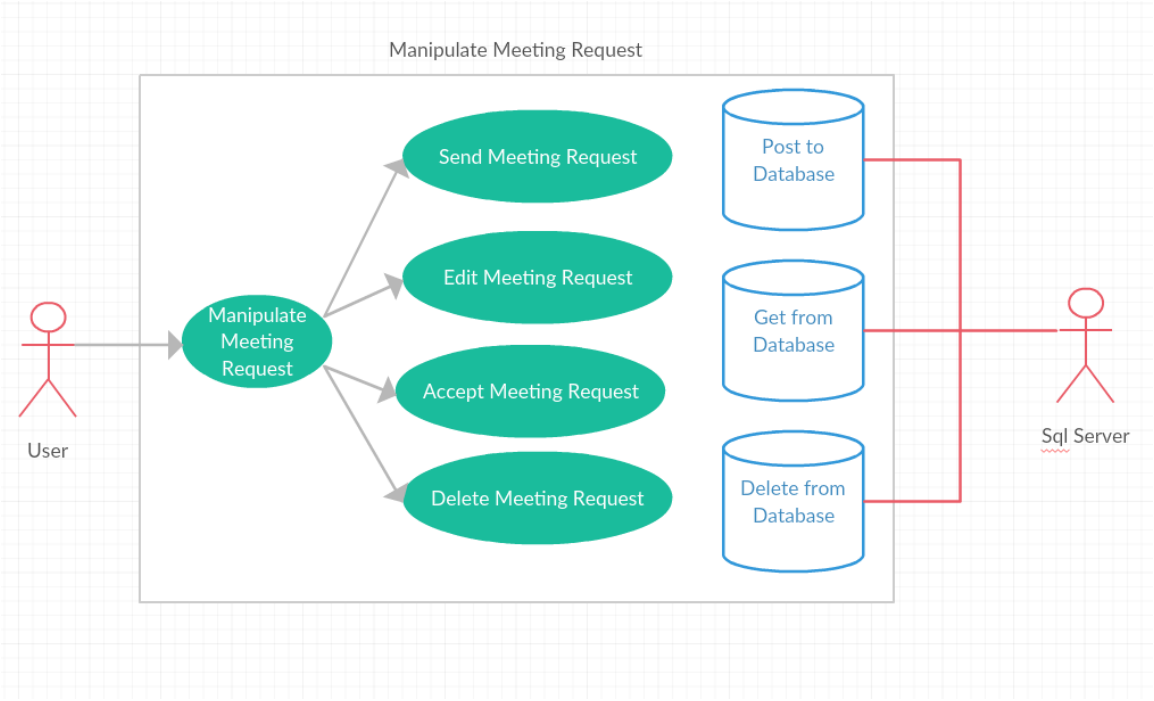
Scope

The scope of this use case is to allow user manipulate meeting request.

Description

This use case describes how to manipulate with meeting request.

Use Case Diagram



Flow Description

Precondition

User must login to his account to access his calendar.

Activation

This use case starts when user chose one of options send, edit, accept, delete meeting request.

Main flow

1. User case begin when user press chosen day at calendar
2. User must enter details of meeting request
3. Submit details
4. System will post data to database

Alternate flow

1. Fields not completed
2. Meeting request not fit into time frame

Termination

Main flow: Meeting request was created.

Alternate flow: Meeting was not created.

Post condition

Main flow: User could see meeting request in calendar.

Alternate flow: User reverted to meeting request form.

2.1.1.7 Requirement 6: Profile

2.1.1.7.1 Description & Priority

This requirement allow user manipulate profile. User are able fully complete details of profile, edit details or delete account.

2.1.1.7.2 Use Case

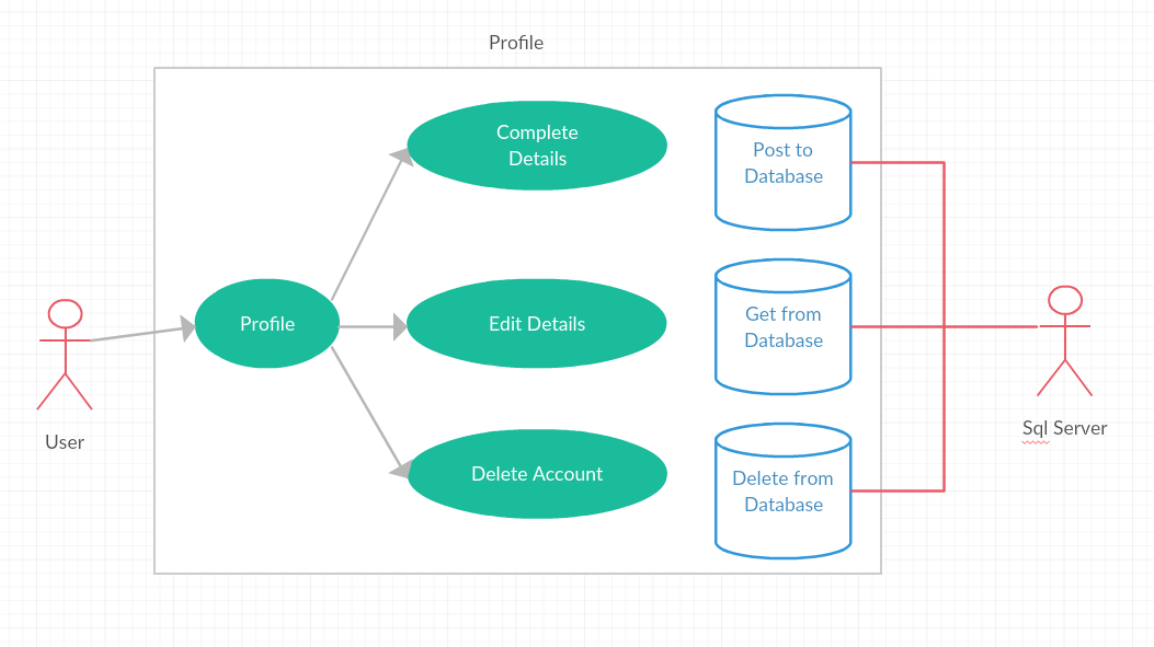
Scope

The scope of this use case is to allow user manipulate his account.

Description

This use case describes how to manipulate own account.

Use Case Diagram



Flow Description

Precondition

User must login to his account to access his profile.

Activation

This use case starts when user account button.

Main flow

1. User case begin when user press button account.
2. User must complete full details of profile
3. Submit details
4. System will post data to database.

Alternate flow

1. Profile must be complete 100%.

Termination

Main flow: Profile is fully completed

Alternate flow: Profile is not fully completed

Post condition

Main flow: User could see details of profile.

Alternate flow: User is required to complete full details.

2.1.1.8 Requirement 7: Messages

2.1.1.8.1 Description & Priority

This requirement allow user send, read, replay, and delete messages.

2.1.1.8.2 Use Case

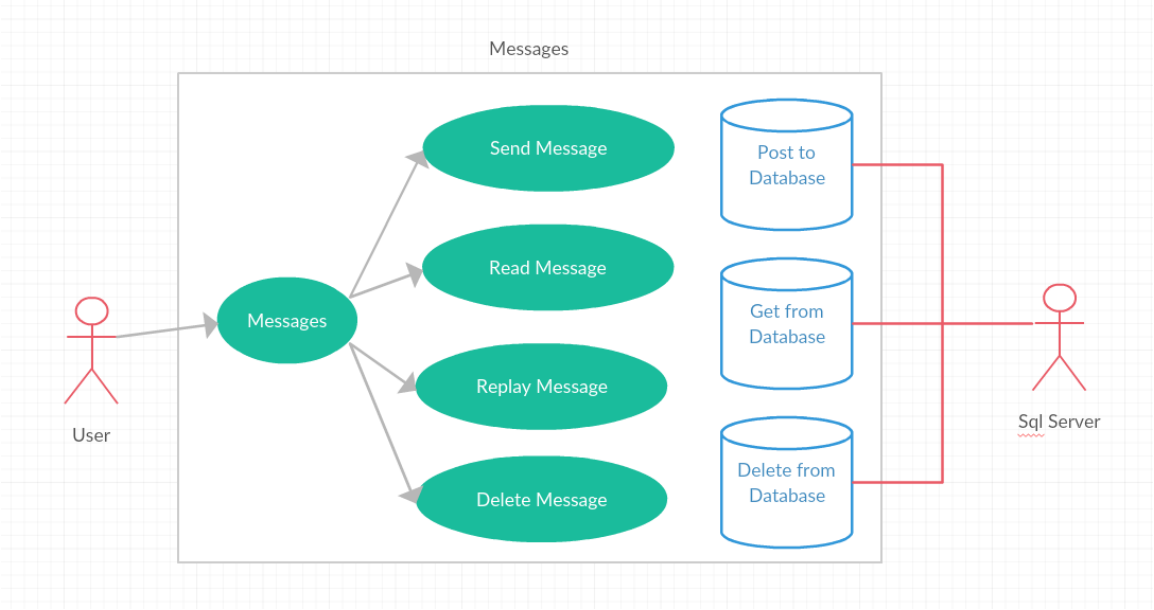
Scope

The scope of this use case is to allow user manipulate messages.

Description

This use case describes how to manipulate with messages.

Use Case Diagram



Flow Description

Precondition

User must login to his account to access messages.

Activation

This use case starts when user press button messages.

Main flow

1. User case begin when users chosen option of messages
2. User must enter details of message
3. Submit details
4. System will post data to database

Alternate flow

1. Field not completed

Termination

Main flow: Message send success, Message replay success, Message delete success.

Alternate flow: Message was not send or replay.

Post condition

Main flow: User could see message history.

Alternate flow: User reverted to message page.

2.1.1.9 Requirement 8: Permission

2.1.1.9.1 Description & Priority

This requirement allow user give permission to other user to access his calendar or profile details.

2.1.1.9.2 Use Case

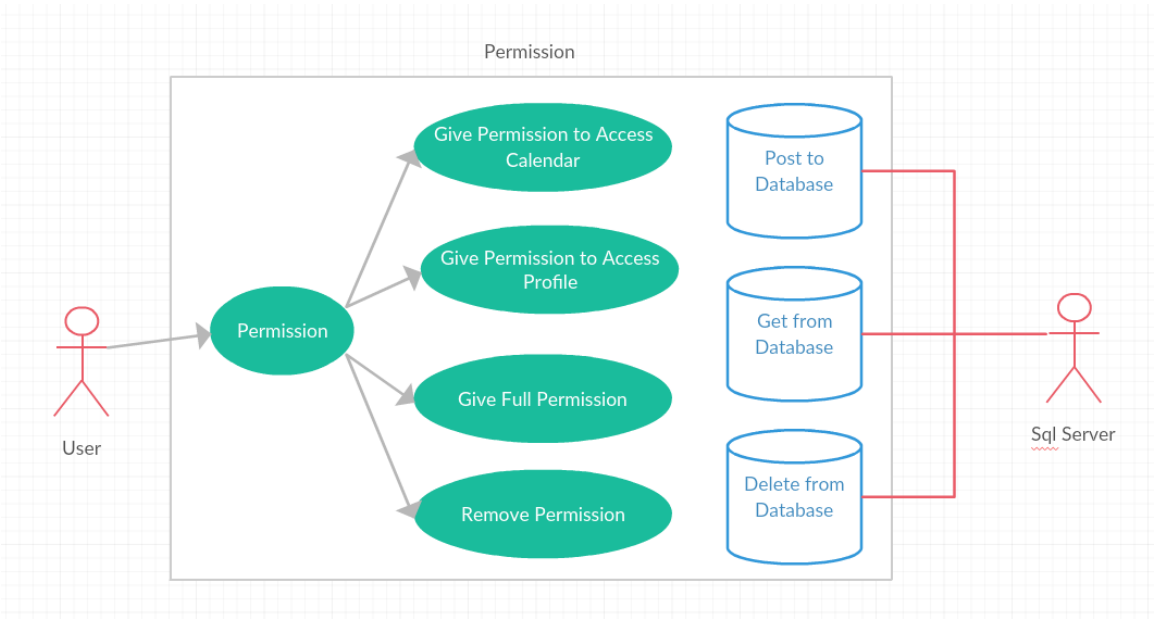
Scope

The scope of this use case is to allow user manipulate permissions.

Description

This use case describes how manipulate with permissions.

Use Case Diagram



Flow Description

Precondition

User must login to his account to access permissions.

Activation

This use case starts when choose one of option calendar permission, profile permission, full access permission, remove permission.

Main flow

1. User case begin when user choose user and press button add permission
2. User must choose what type of permission hi want to apply
3. Submit chosen permission
4. System will post data to database

Alternate flow

1. User already has permission

Termination

Main flow: Permission was applied.

Alternate flow: Permission already applied.

Post condition

Main flow: User can access your profile or calendar.

Alternate flow: User reverted to permission form.

2.1.2 Non-Functional requirements

2.1.2.1 Performance/Response time requirement

The application should be easy to use. Performance will be a major aspect in the overall application. The application should be capable of being opened on any browser or mobile devices without any response time issues or layout issues.

2.1.2.2 Availability requirement

The application should be available at all web browser or all size device.

2.1.2.3 Recover requirement

The application should be able to recover from an error quickly. In the event of a server failure, it should be possible to quickly restart the server.

2.1.2.4 Robustness requirement

The application should be able to appropriately react to real-time errors during execution. For example, if a user attempts to enter an invalid character into a text-field, the app should display an error to the user to prevent an invalid input into the system.

2.1.2.5 Security requirement

User's emails and passwords will be encrypted and all data will be sanitized to prevent from SQL injections.

2.1.2.6 Reliability requirement

- Availability - Percentage of time available.
- Accuracy - The user input should be accurately maintained.
- Security Considerations - The application should ensure the privacy of user data.

2.1.2.7 Maintainability requirement

- Any problem that occurs from use of the application will be easily identifiable the first time it occurs.
- The application can then be easily modified without causing errors in other parts of the application.

2.1.2.8 Portability requirement

Portability requirements will not be an issue because this application will be online based.

The application will be easily and quickly ported to new device because it will be size response.

- Programming Language – PHP, JavaScript, HTML, and CSS.
- Browsers - Available across all browsers.

2.1.2.9 Extendibility requirement

The application will be extensible. It will be self-build application and more user use it more extensions will be involve.

2.1.2.10 Reusability requirement

Reusability requirement will be user should login to use application. If user close browser then session will end and user will be logout automatically.

2.1.3 Data requirements

The application will use relation database and required tables to store data.

- User
- Company
- Request
- Meeting
- Calendar
- Staging calendar

2.1.4 User requirements

- Separate calendar for each company
- Ability create profile
- Ability upload image
- Ability to share free/busy time with others
- Ability to send/edit/accept/delete meeting request
- Notification of received request
- Responsive web design

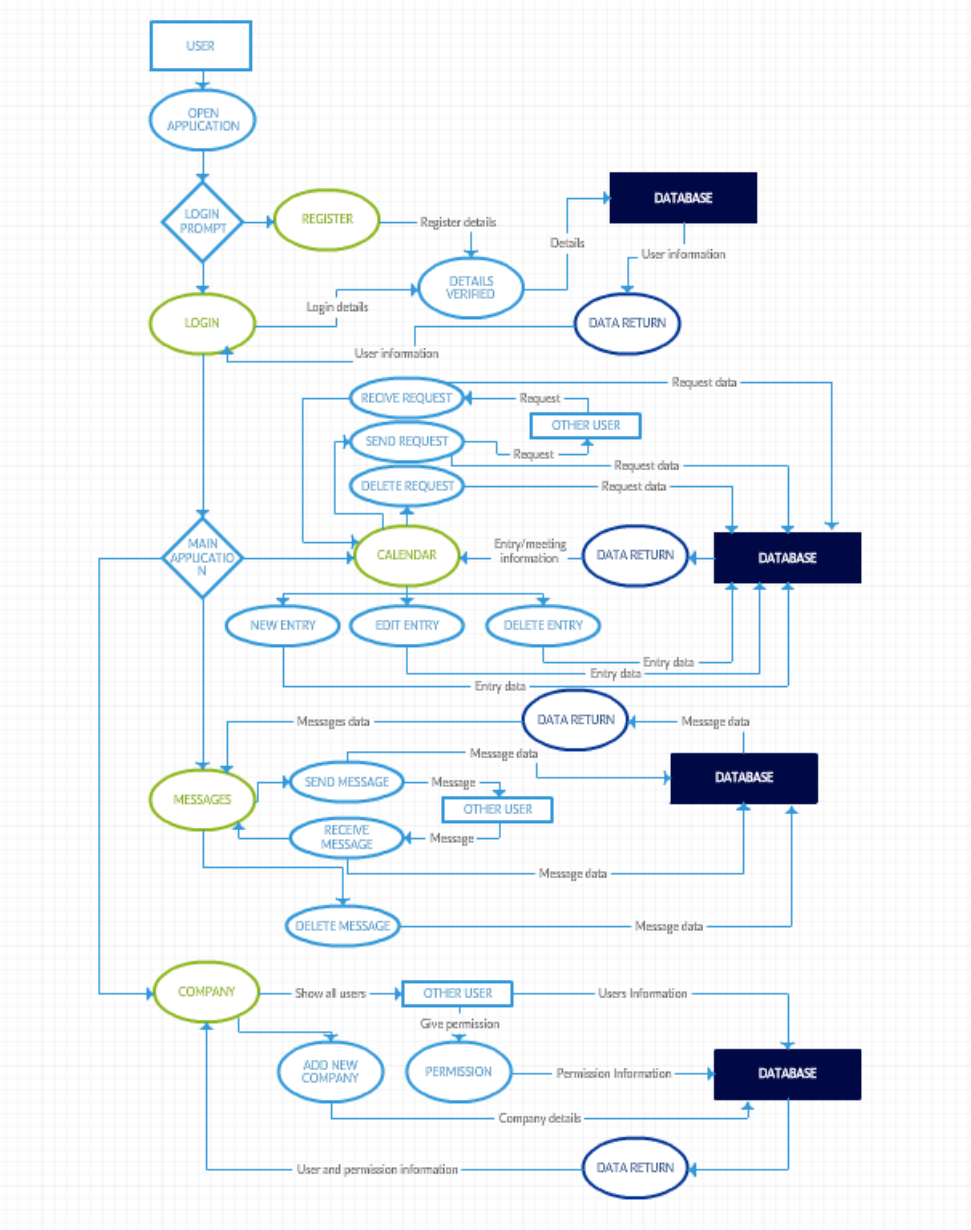
2.1.5 Environmental requirements

The application will require an active internet connection and could be run at any device.

2.1.6 Usability requirements

Application welcome page could be accessed by everybody. For new user there is option create new account. To get full access of application user has to login into his account.

2.2 Design and Architecture



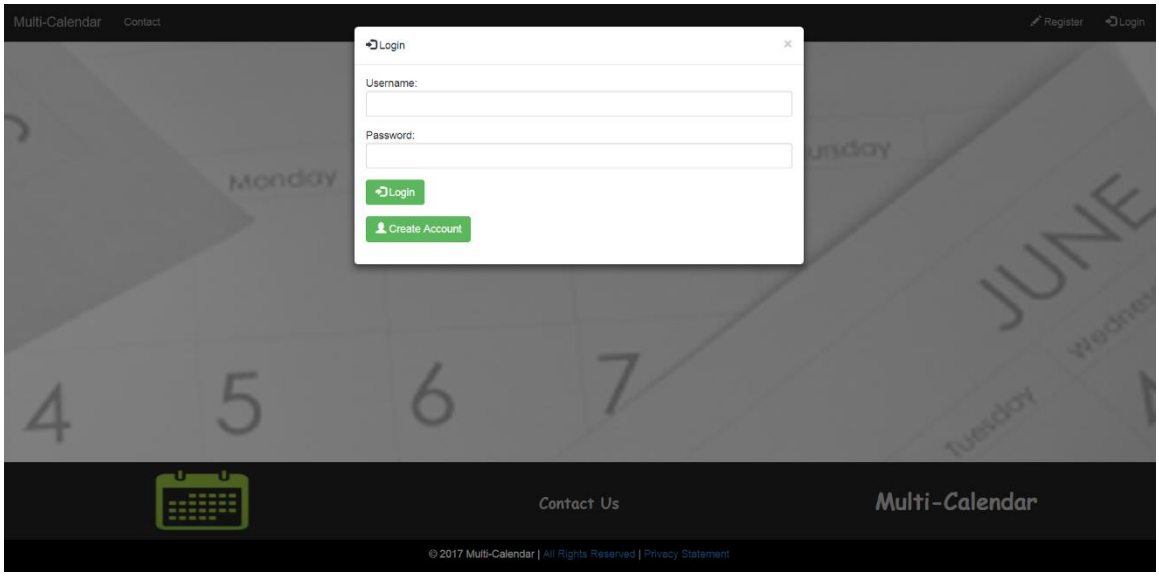
2.2.1 Implementation

Describe the main algorithms/classes/functions used in the code. Consider to show and explain interesting code snippets where appropriate.

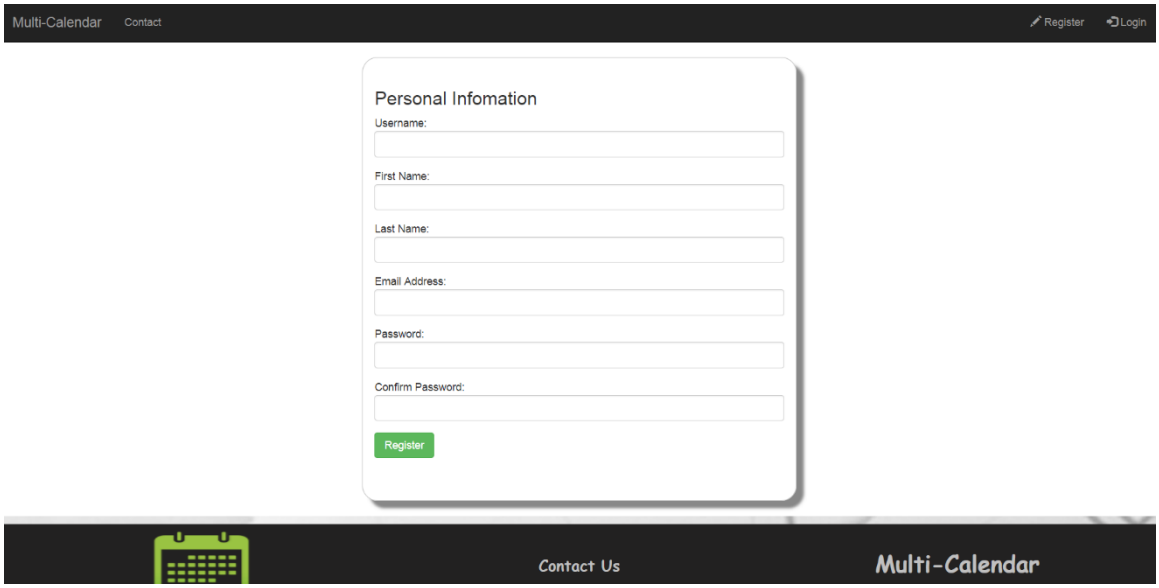
2.2.2 Graphical User Interface (GUI) Layout

Welcome page

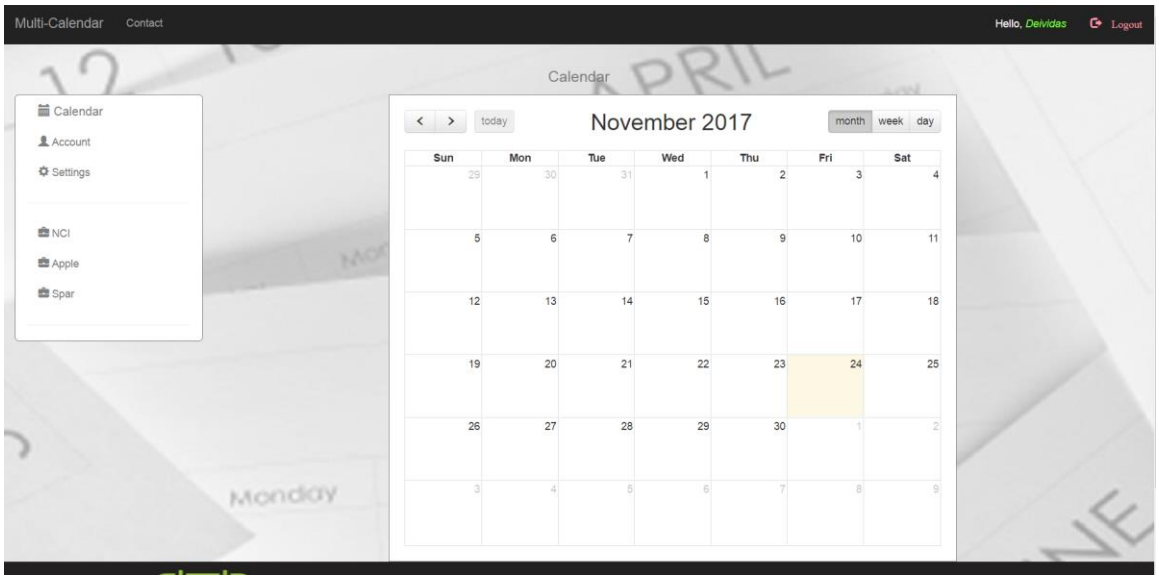
Login Modal



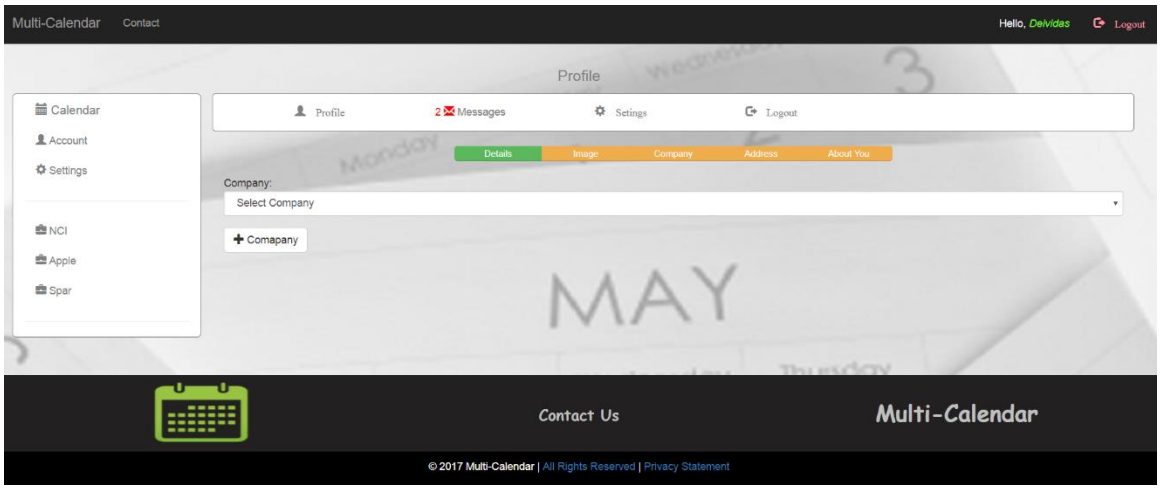
Register page



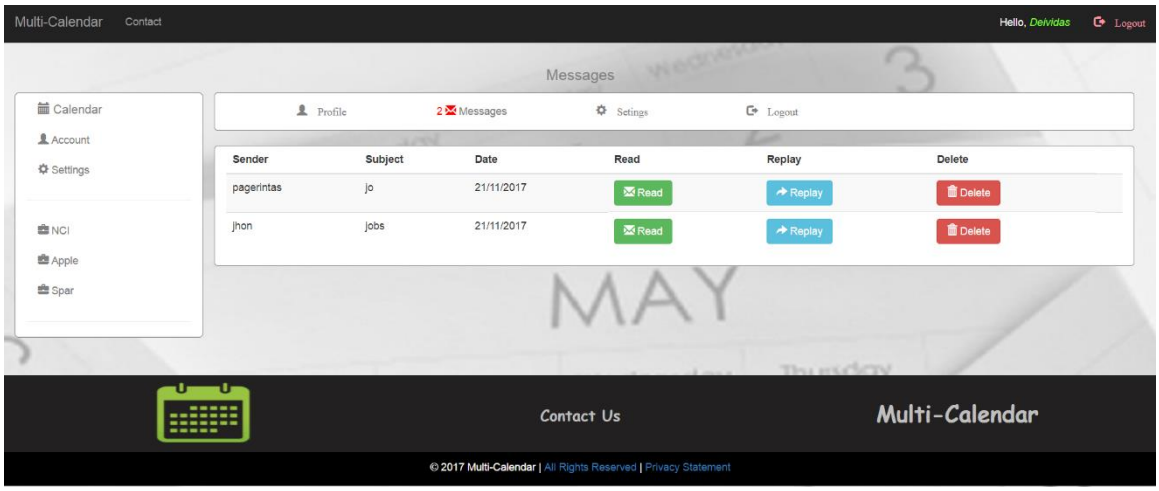
Calendar page



Account page



Messages pages



2.2.3 Testing

Describe any testing tools, test plans and test specifications used in the project

2.2.4 Customer testing

Provide evidence for and results of customer testing. This may include ratings or quotes from the customer.

2.2.5 Evaluation

How was the system evaluated and what are the results? In many cases this will include usage data and user feedback. It may also include performance evaluations, scalability, correctness, etc. depending on the focus of the project.

Quantative results may be reported in tables or figures. Note that tables have their caption above the table and need to be cross referenced in the text (see **Error! Reference source not found.**). In many cases, tables are better to read if you skip the vertical lines.

Table 1: Performance with and without caching

	$N_{without}$	N_{with}	$Std.\text{-}Deviation_{with}$	$Std.\text{-}Deviation_{without}$	p
Records	100	200	2.54	3.97	.002
Data (GB)	100	200	2.54	3.97	.002
Speed	100	200	2.54	3.97	.002

Figures have their caption below the figure as shown in **Error! Reference source not found.**. Make sure that if you use colour, the figure is still readable when printed in black & white, e.g., by using additional symbols, patterns, etc.

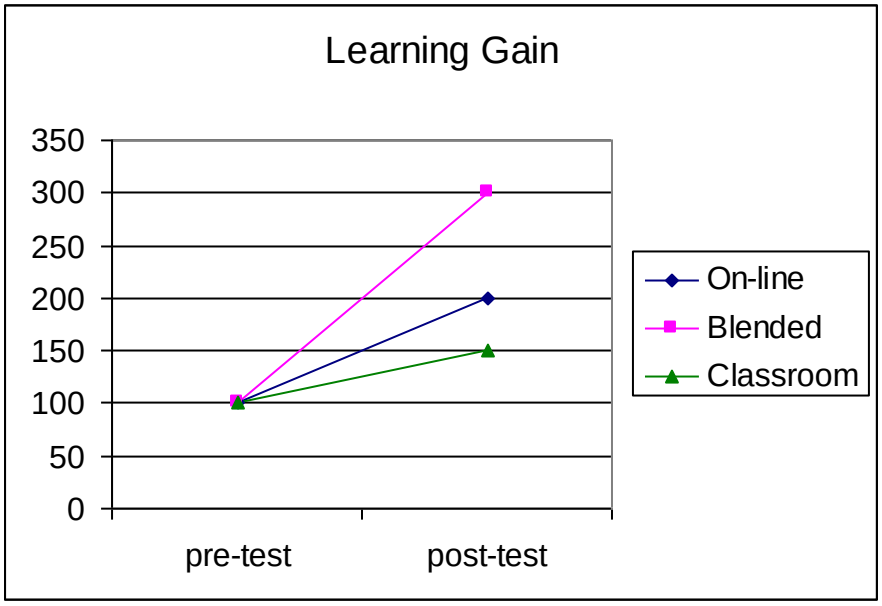


Figure 1: Learning gain across different experimental groups

3 Conclusions

Describe the advantages/disadvantages, opportunities and limits of the project.

4 Further development or research

With more resources, where could the results of this project lead to?

5 References

It is recommended that students use the APA, Berkeley, Harvard or other internationally approved style. Here is an example of the APA citation style:

Wilcox, R. V. (1991). Shifting roles and synthetic women in Star Trek: The Next Generation. *Studies in Popular Culture, 13*(2), 53-65.

In the text this article can be cited as “Wilcox (1991)” or “(Wilkox, 1991)”.

References to web sites must include the access dates.

The library provides a study guide on Harvard style referencing.

6 Appendix

6.1 Project Proposal

6.1.1 Objectives

Objective of this project is to develop fully functional multi-calendar web application. Target of users will be people who doing multiple jobs. For example if person is working in Company A, Company B, and Company C then he has to maintain three different calendars. This application will allow user to create many calendars he want in one place. With multi-calendar user will be able to share his free and busy time, check any other user availability, arrange meeting.

6.1.2 Background

Calendar

Main purpose of this application to allow user create multiple calendars for different companies. Also add business partners for each company. Register user could check free and busy time of any user without permission but he should be register in same company. By following this calendar user could easer to plan his day.

Meeting

User is able to create meeting with business partner or multiple business partners. If user send request of meeting then business partner have to accept or reject. By accepting request he will automatically edit his calendar with one extra meeting. If user want to send request of meeting for multiple business partners then he is able to choose two type of meeting. First will be all of group members should accept request other way meeting will be cancelled. Second type one of group members should accept request.

Permission

Owner of calendar is able to give permission for any business partner. If user has permission access business partner calendar he is able to see content of meetings.

6.1.3 Technical Approach

I will make prototype of application how it should look then I will add required functionality. Without registration user is not able to use this system. Ensure there is no duplicated users in database. Logged in user should see correct details. Security of user details.

6.1.4 Special resources required

At this stage I have no require special resource because I will use Cloud9 for coding and there are everything integrated.

6.1.5 Technical Details

Languages: PHP, HTML, JavaScript.

Library: Bootstrap, JQuery.

I will use Cloud9 development environment it has integrated Apache and MySQL servers, also allow me do my coding at any machine.

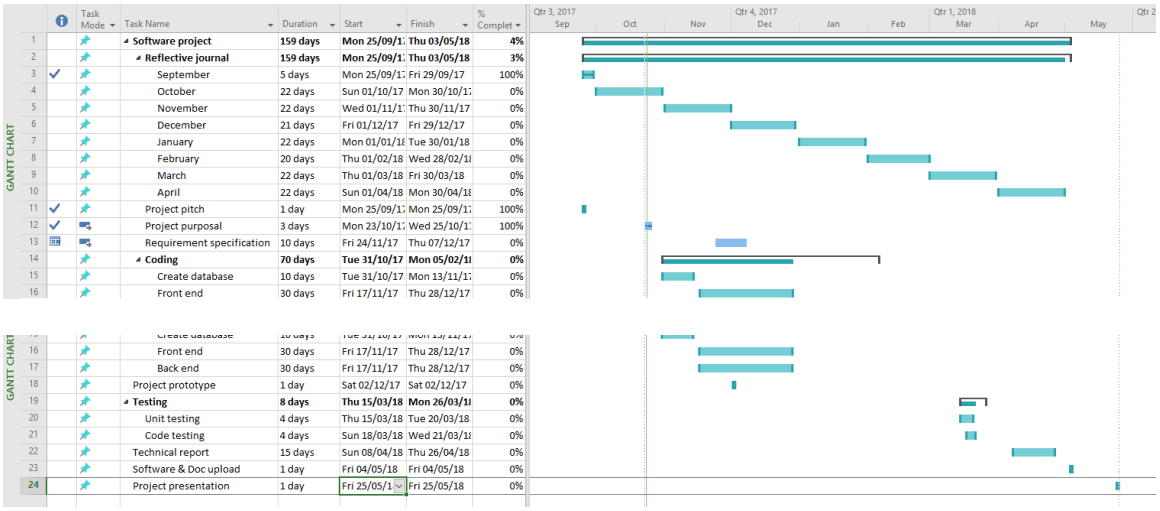
To store data I will use PHPMyAdmin.

To deploy project I will use Microsoft Azure.

6.1.6 Evaluation

Testing I will do myself as part of development process. GUI testing - this application should run at any device as desktop, laptop, mobile, iPad, and tablet. Database testing – this application will use multiple datasets and all data has to be save I correct tables. Security testing – to ensure application is fully secure. When I have this application ready I will ask friends to do some testing and if they will find some debugs or error I will make changes.

6.2 Project Plan



6.3 Monthly Journals

6.3.1 September

Student name: Deividas Sevcenko

Program: BSc in Computing

Month: September

This month, I was thinking about project idea. My idea for this project is online auction for Ireland market. I have chosen use PHP language for this project. First I did some planning how everything should look and what functionality I should implement to make fully functional web application. I would like to make this application make fully automated for example owner of application shouldn't spend a lot of time to deal with customers or to arrange live auctions. Auctions will be chosen by user when it should start. Buyer and seller will communicate between each other. For example if item was bought buyer will get email with details of seller and seller will get email with buyer details. It will be set some rules about delivery. Also it will be 10% commissions from seller.

I have realized that I need to have supervisor. Next month I will meet my supervisor and I will discuss about my idea and it will be fine then I will continue with coding.

Supervisor Meetings

Date of Meeting:

Items discussed:

Action Items:

6.3.2 October

Student name: Deividas Sevcenko

Program: BSc in Computing

Month: October

On beginning of October I am due to present my idea in order to receive approval. My idea was rejected because it is to scale and a lot of already exist. Next step was to choose one of ideas in the list. I have chosen one of my supervisor idea it is multiple calendar. To choose idea was easy but make it real not so easy. I start my planning for this project. First I start planning how it should look. What functionality I need to involve to make it work properly. After I have done project proposal I start some coding. First I have created project in Cloud9 and created database connection with database. I have created database in PHPMyAdmin and named it multi-calendar. I have started some coding by creating required pages. Next month I will finish project requirement specification document and will continue with coding. By during my coding step by step I will be sure what tables I should create in database.

Supervisor Meetings

Date of Meeting: 10/20/2017

Items discussed: auction idea wasn't accepted, I have to choose idea from list

Action Items: finish project proposal, start project requirement specification

6.4 Other Material Used

Any other reference material used in the project for example evaluation surveys etc.