CEBU INSTITUTE OF TECHNOLOGY UNIVERSITY

COLLEGE OF COMPUTER STUDIES

CIT-U EVENTS CALENDAR

Team Member:

Jan Michael G. Ragasajo

Chris Ian L. Berbo

Joshua C. Bacarisas

John Mike H. Judan

Mel Patrick Lecciones

Adviser:

Mr. Mario Silvano

ABSTRACT:

This is software CIT-U Events Calendar is intended for faculty, staff, and students specifically the new students. CIT-U Calendar Events App gives the students, staff, and faculty the information in our school. Its main purpose is to provide vital information for CIT-U's students, staff, and faculty members about the school events, memos, holidays, announcements and schedules about examinations.

Software Requirements Specifications

for

CIT-U Events Calendar

Signature

Jan Michael G. Ragasajo	-	Developer / UI Designer
Chris Ian L. Berbo	-	Team Leader / Tester
Joshua C. Bacarisas	-	Developer / Tester
John Mike H. Judan	-	Developer / Analyst
Mel Patrick Lecciones	-	Documentation / Tester

Change History

Name	Date	Changes
Preliminary Version	August 16, 2014	**Basic design of the website is changed**
Secondary Version	September 18, 2014	**Mobile version is removed, new UI is added, Super admin functions are added and holiday events added.**
Tertiary Version	September 27, 2014	**Date range function added, dummy creation removed, new modal added, Recolors done to the main page**

Preface

This is the software requirements specifications (SRS) document for the CIT-U Events Calendar In particular, the document details the features, functions and software specifications of this Events. The intended users of the SRS are the faculty, staff, and students specifically the new students.

Table of Contents

List of Figures

1. Introduction

1.1. Purpose

Develop a CIT-U Calendar Events App allowing the students, staff, and faculty the information in our school. Our main purpose of this project is to provide vital information for CIT-U's students, staff, and faculty members about the school events, memos, holidays, announcements and schedules about examinations.

1.2. Scope

The scope of this document is to describe the over all description of the project itself. And to define whether it has any constraints or restrictions

1.3. Definitions, Acronyms and Abbreviations

1.4. References

- www.google.com
- IEEE 12207.2-1997 Industry Implementation of International Standard ISO/IEC 12207: 1995 (ISO/IEC 12207)Standard for Information Technology - Software Life Cycle Processes - Implementation Considerations
- IEEE Std. 830-1998 IEEE Recommended Practice for Software Requirements Specifications
- [IEEE] The applicable IEEE standards are published in "IEEE Standards Collection," 2001 edition.

1.5. Overview

2. Overall Description

2.1. Product perspective

- This CIT-U Events Calendar will be used as a guideline tool for students.
- This CIT-U Events Calendar will be available to almost all of the mobile devices.

2.2. Product functions

- User View
    - Select Department
    - View events
    - Recall current date

- Admin
    - Add
    - Update
    - Delete
    - Edit
    - Date range function

- Super Admin
    - Manage admins
    - Activate admin accounts
    - Block/unblock admin accounts
    - Decline admin accounts
    - Add dept.
    - Edit dept.
    - Edit profile

2.3. User characteristics

- Students, faculty and staff will be aware of the events.
    - Example for the event is intramurals. The students will know the exact details and time of the said events.

2.4. Constraints

NONE

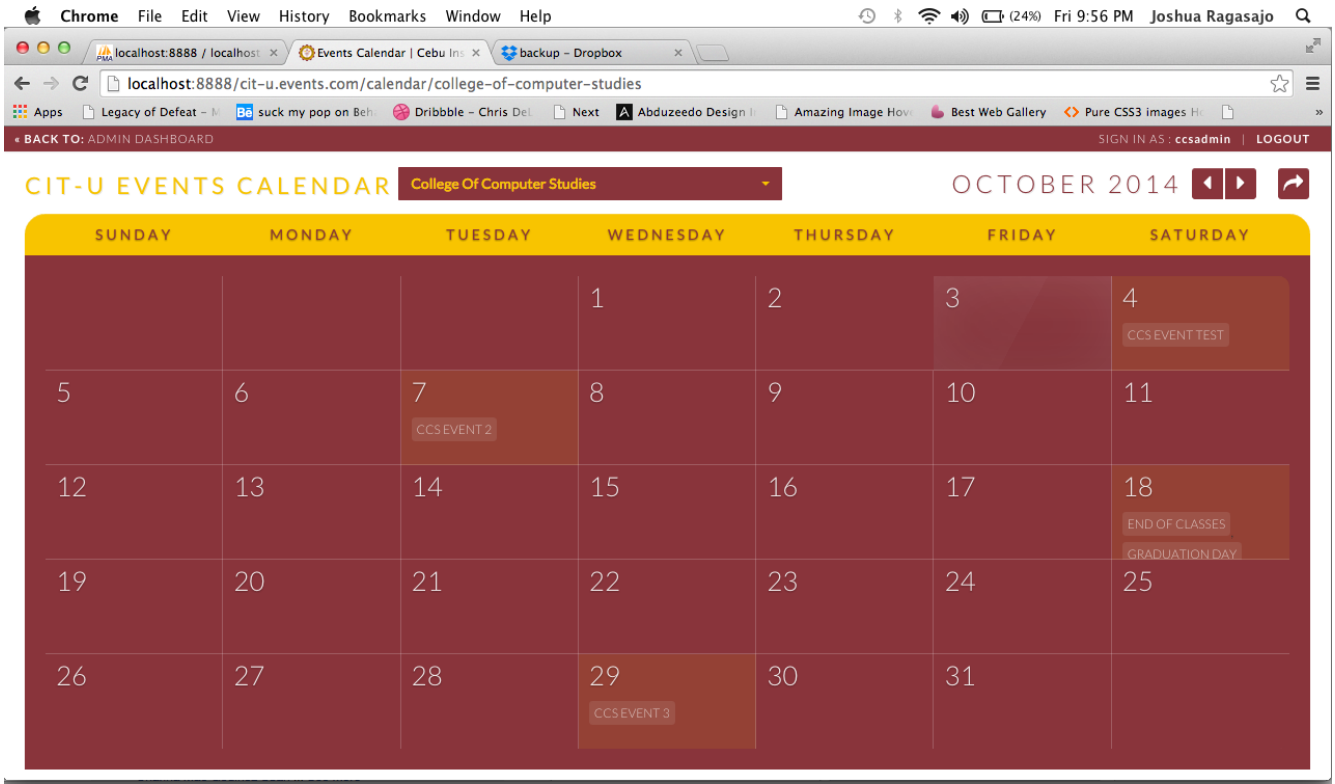
2.4. Assumptions and dependencies

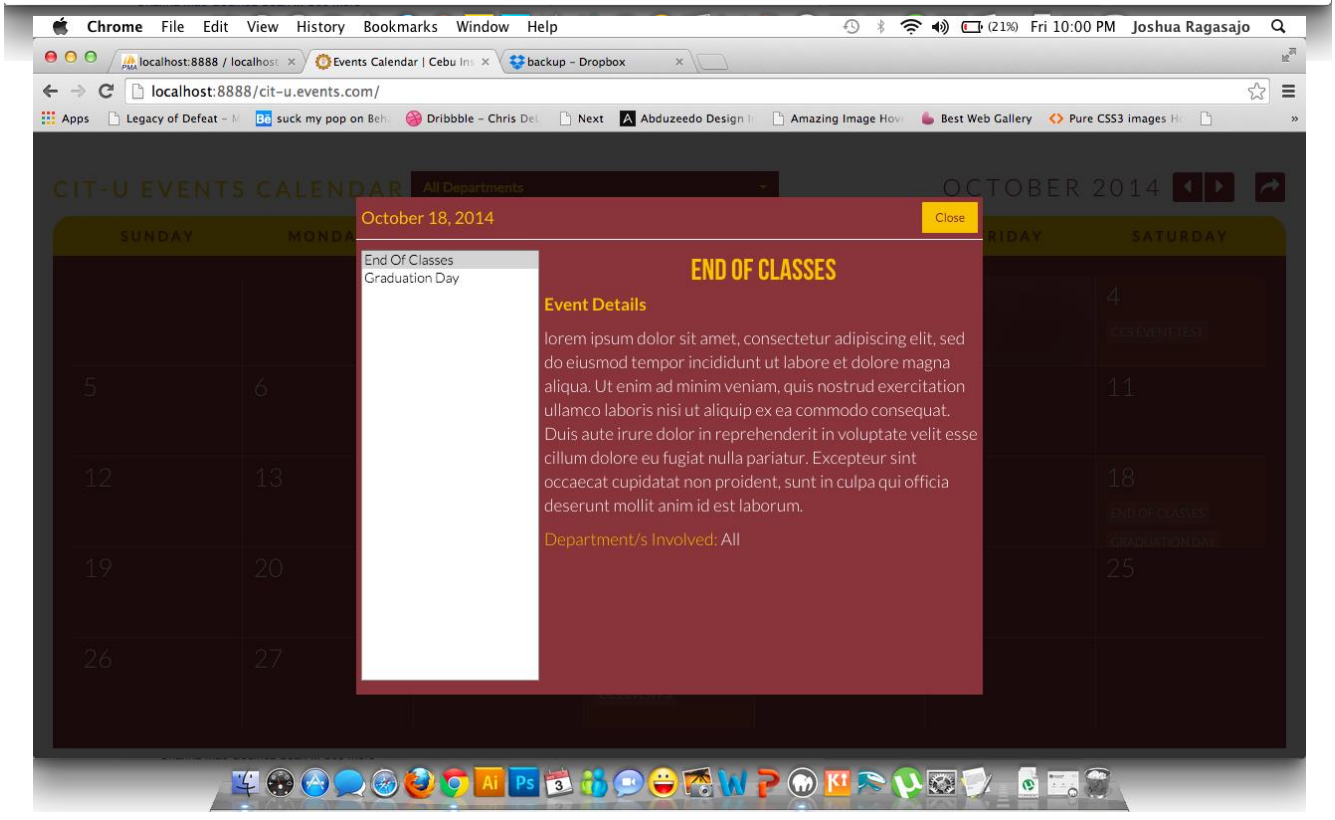
NONE

3. Specific Requirements

3.1. External interface requirements

3.1.1. User interfaces





3.1.2. Hardware interfaces

- None: This is a web and mobile application
- Mobile phone
- PC or MAC

3.1.3. Software interfaces

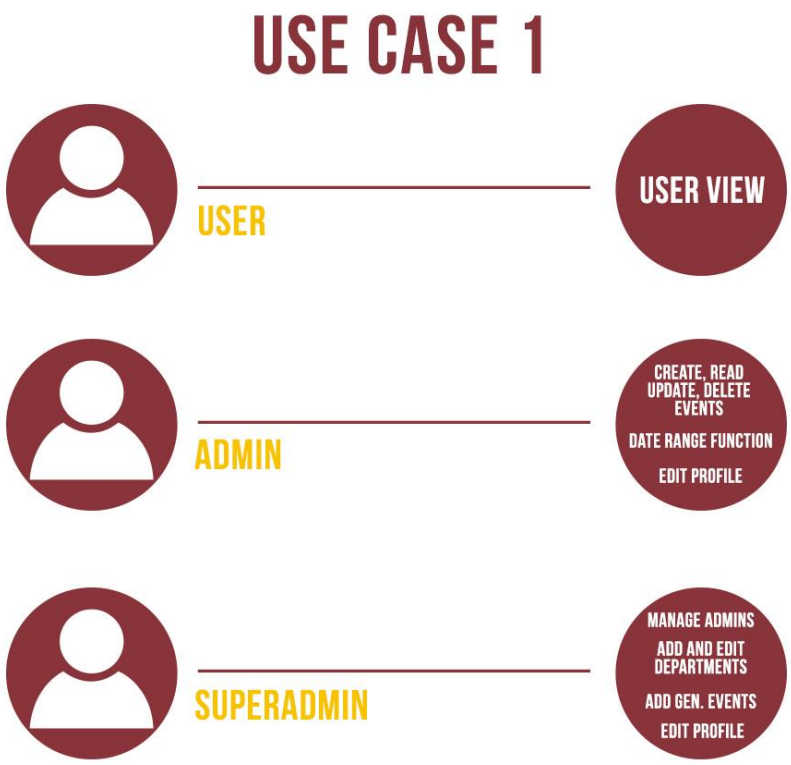
- Database

3.1.4. Communications interfaces

NONE

3.2. Functional requirements

3.2.1. Use case 1



User View:

- User can view events.

Admin View:

- Admin can delete, add, and update. Can query specified events using the date range function. Manage their profile settings.

Super admin View:

- Super admin can manage admins(Approve or decline requests, block or unblock admins). It can also add departments, generate dummy user (for testing purposes), add general events which applies to all departments and it can also manage his / her profile settings.

3.3. Performance Requirements

- Any iOS device
- Minimum of android version 2.2 (froyo)
- Windows XP or higher PC or Laptop

3.4. Design constraints

NONE

3.5. Software system attributes

NONE

3.6. Other requirements

NONE

4. Appendixes

5. Index

Software Project Management Plan (SPMP)

For

CIT-U Events

CCS417 Software Project

October 4, 2014

• Signature

Contributor	Assignment	Signature
Jan Michael G. Ragasajo	Back – End Developer / UI Designer	
Chris Ian L. Berbo	Team Leader / Tester	
Joshua C. Bacarisas	Mobile Developer / Tester	
John Mike H. Judan	Mobile Developer / Analyst	
Mel Patrick B. Lecciones	Documentation / Tester	

• Change History

Name	Date	Reason For Changes
Preliminary Version	August 16, 2014	**Basic design of the website is changed**
Secondary Version	September 18, 2014	**Mobile version is removed, new UI is added, Super admin functions are added and holiday events added.**
Tertiary Version	September 27, 2014	**Date range function added, dummy creation removed, new modal added, Recolors done to the main page**

• Preface

This document contains the Software Project Management Plan (SPMP) of Cebu Institute of Technology University Events Calendar. The main goal of this is to develop a website that will serve as the go to site for CIT-U students or members when looking for events, announcements, holidays etc. .

This document has been prepared in accordance with the IEEE Std 830-1998, IEEE Recommended Practice for Software Project Management Plan [IEEE 1998].

• Table of Contents

• Overview

5.1. Project Summary

5.1.1. Purpose, scope and objectives

Our main purpose of this project is to provide vital information for CIT-U's students, staff, and faculty members about the school events, memos, holidays, announcements and schedules about examinations.

The objectives of the project are as follows:

1. Complete the project by the project due date

2. Fulfill all stated requirements in the SRS

3.The team will work together to complete the project

5.1.2. Assumptions and constraints

The project has been planned with the following assumptions:

1. This project will deliver only a software component.
2. The project is still under development.
3. The product must be reliable.
4. The product must be user-friendly.

This has also been planned with the following constraints:

- Time

  The whole project is only good for 14 weeks.

- Staff

  Our project adviser will act as the Project Consultant.

  The development team is composed of five (5) members. The members are expected to initiate research on areas needed for the development.

- Maintenance

  The software will be maintained by the software developers.

5.1.3. Project deliverables

As part of the project, the project team will deliver the following documents to the client:

1. **Project Proposal**
   -Contains objective of the project and an introduction.

2. **Software Requirements Specification(SRS)**
   -Details what is required of the system by the Client. It contains both functional and non-functional requirements.

3. **Software Project Management Plan(SPMP)**
   -Outlines the project and people involved and details the processes and guidelines to be followed throughout the duration of the project.

4. **Software Design Document(SDD)**
   -Contains an in-depth design of the software, including use-case, collaboration, sequence and class diagrams.

5. **Software Test Document(STD)**
   -Details the types of tests to be carried out on the system to ensure the system meets requirements and maintains integrity.

6. **Software Output**

5.1.4. Schedule and budget summary

The project has an allocated budget for each member's compensation rate of $... per hour. Miscellaneous expenses will be each member's responsibility. An estimate of …. man-hours will be needed for this project, with a total estimated budget of …..

The target schedule for each project milestone is shown in the table below:

Milestone	Date Deadline
Project Initiation	June 28,2014
Requirements Specification	July 26, 2014
Management Plan	August 2, 2014
Software Design	August 9, 2014
Software Testing	August 16, 2014
Working Output	Whole month of September
Presentation of Output	3rd week of September

[Table 1.1.4] Schedule and Allocation

5.2. Evolution of plan

Version	Description of Version	Date Expected

Initial	Rough draft created for initial submission and review comments.	August 16,2014
Final	Final Submission of Documents	September 27,2014

[Table 1.2] Evolution Plan

• References

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.* 6.
IEEE Computer Society, 1998.

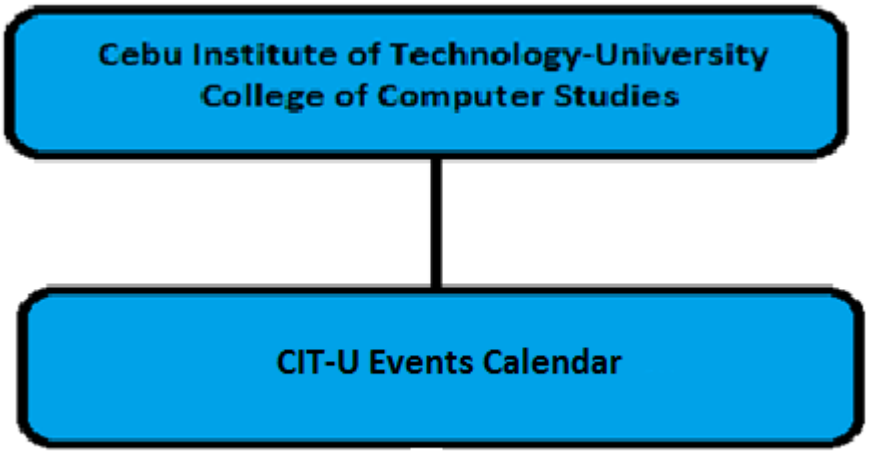
• Definitions

CIT-U	Cebu Institute of Technology- University
CCS	College of Computer Studies
SP	Software Project
SRS	Software Requirements Specification
SPMP	Software Project Management Plan
DB	Database
GUI	Graphical User Interface

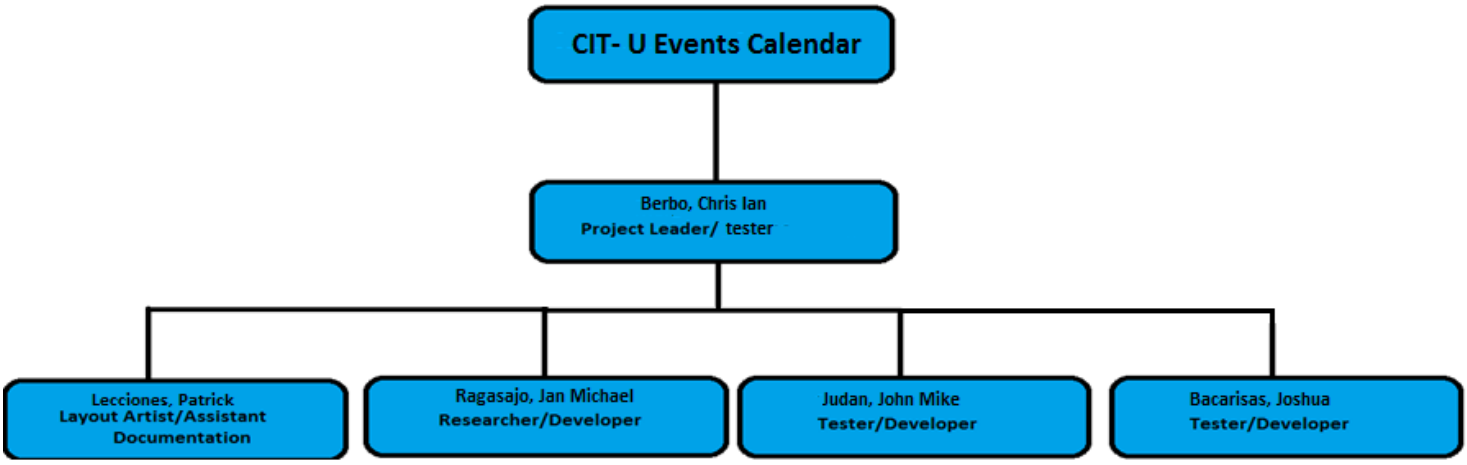
Project organization

This clause of the SPMP shall identify interfaces to organizational entities external to the project; describe the project's internal organizational structure; and define roles and responsibilities for the project.

6.1. External structure



6.2. Internal structure



4.3. Roles and responsibilities

Role	Responsibilities
Project Manager	Supervises the team throughout the development process. Validates all deliverables and reviews all documentation.
Team Lead	Leads the development Assures that each deliverables is on its milestones. Monitors member's task.
Developer	Responsible for hard coding. Presents ideas for system improvements, including cost proposals Working closely with clients, designers and staff Maintaining the systems once they are up and running
Layout Artist	Develops its GUI and other necessary designs for the Subsystem.
Tester	Tests all parts of the system after development stage.
Researcher	Responsible for related studies that might help in developing this project

Documentation	Responsible for any documentation needed in this development

[Table 4.3-A] Roles and Responsibilities

Role	*Person Responsible*
Team Lead *Tester*	*Berbo Chris Ian*
Layout Artist *Developer*	*Ragasajo Jan Michael*
Researcher *Documentation*	*Lecciones, Mel Patrick*
Tester *Developer*	*Judan, John Mike*
Tester	*Bacarisas Joshua*

[Table 4.3-B] Role Allocation

Managerial process plans

This section will stipulate the Project Management Process of the Sales and Inventory Control System. It will show the detailed planning of the project for the group to deliver it in time. All members work hand in hand to achieve the project's objectives.

6.3. Start-up plan

6.3.1. Estimation plan

The estimated total development time of the system is 13 weeks, which started on June 28, 2014. It will be divided into 6 different deliverables based on the phases of Software Engineering. Each deliverable will have its documentation and will be counted as milestones.

The development is properly scheduled using the Ms Project for the Work Breakdown Structure. The MS Project will be used for us to keep track on what is scheduled, most especially the milestones, and also on the resource allocation of the project.

Estimated total cost for the project will be 40,000P. Figures were obtained based on the 350P rate or each member of the team. Overtime pays are not included in the estimation, since members must target to finish each task on time. If in case of unfinished task or an additional amount of time needed to be allocated, it's the developer's responsibility to go on overtime just to beat the deadline. In this estimation, we are using the Parkinson's Law, wherein each member's work expands to fill the time available.

6.3.2. Staffing plan

Name	Affiliation to the Project	Most Available Schedule
Berbo, Chris Ian	CIT-U CCS Student/Team Leader, Tester	Monday, Tuesday, Wednesday, Friday and Saturday
Ragasajo, Jan Michael	CIT-U CCS Student/Layout Artist, Developer	Monday, Tuesday, Wednesday, Friday and Saturday
Lecciones, Patrick	CIT-U CCS Student/ Researcher Tester	Monday, Tuesday, Wednesday, Friday and Saturday
Judan, John Mike	CIT-U CCS Student/Tester, Developer	Monday, Tuesday, Wednesday, Friday and Saturday

Bacarisas, Joshua	CIT-U CCS Student/Tester, Researcher	Tuesday, Thursday, Friday , Saturday

[Table 5.1.2] Staff Plan

5.1.3. Resource acquisition plan

Software resources needed for the project are all open-source and are readily available anytime. Each member has their own laptop/desktop computer for the development of the project.

This system will be deployed on client serve.

5.1.4. Project staff training plan

There will be no training provided to the project team. Each member is equipped with the necessary knowledge needed to develop the system. Any additional knowledge needed for the development will be on the member's initiative to research and learn that area.

6.4. Work plan

6.4.1. Work activities

Task Name	Duration	Start	Finish	Predecessors	Resource Names
Software Development Plan for Sales and Inventory Control System	**64 days**	**Sat 6/28/14**	**Sat 9/20/14**		
Proposal Period	**7 days**	**Sat 6/28/14**	**Sat 7/5/14**		
Determine project scope	4 hrs	Mon 6/30/14	Mon 6/30/14		Project Team
Research Relevant to the Project	1 day	Mon 6/30/14	Tue 7/1/14	2	Project Team
Define preliminary resources	1 day	Tue 7/1/14	Wed 7/2/14	3	Project Team
Brainstorming for Proposal	1 day	Wed 7/2/14	Thu 7/3/14	4	Project Team
Scope complete	0 days	Thu 7/3/14	Thu 7/3/14	5	
Analysis/Software Requirements	**18 days**	**Thu 7/3/14**	**Mon 7/28/14**		
Meeting for Software Specifications	1.5 days	Thu 7/3/14	Fri 7/4/14	6	Project Team
Draft preliminary software specifications	3 days	Mon 7/14/14	Wed 7/16/14		Project Team
Review Preliminary Software Specifications	2 days	Thu 7/17/14	Fri 7/18/14	9	Project Team
Present Software Specification Requirements	4 hrs	Sat 7/26/14	Sat 7/26/14		Project Manager,Analyst
Incorporate feedback on software specifications	1 day	Sat 7/26/14	Mon 7/28/14	11	Analyst,Project Manager
Software Project Management Plan	**8 days**	**Mon 7/28/14**	**Tue 8/5/14**		
Meeting with team for Software Management Plan	1 day	Mon 7/28/14	Mon 7/28/14		Project Team
Review Software Requirements	1 day	Tue 8/5/14	Tue 8/5/14	14	Project Team
Review Management Plan	1 day	Thu 7/31/14	Thu 7/31/14		Project Team
Present Software Management Plan	1 day	Sat 8/2/14	Sat 8/2/14	16	Analyst,Project Manag
Incorporate feedback	0.5 days	Sat 8/2/14	Sat 8/2/14		Analyst,Project Manag
Software Design	**18 days**	**Mon 8/4/14**	**Tue 8/26/14**		
Meeting with the team for Software Design	1 day	Mon 8/4/14	Mon 8/4/14		Project Team
Review Specific Requirements	1 day	Tue 8/5/14	Tue 8/5/14	20	Developer
Develop Functional Requirements	1 day	Wed 8/6/14	Wed 8/6/14	21	Developer
Review Functional Requirements	15 days	Thu 8/7/14	Tue 8/26/14	22	Developer
Present Software Design	1 day	Sat 8/9/14	Sat 8/9/14		Project Manager,Analy
Incorporate Feedback	0 days	Sat 8/9/14	Sat 8/9/14	24	Project Team
Software Development	**7 days**	**Sun 8/10/14**	**Sat 8/16/14**		
Meeting for task distribution	1 day	Mon 8/11/14	Mon 8/11/14		Project Team
Develop System Code	4 days	Mon 8/11/14	Thu 8/14/14		Developer
Preliminary Debugging	1 day?	Thu 8/14/14	Thu 8/14/14		Developer
Testing	**7 days**	**Tue 8/12/14**	**Tue 8/19/14**		
Develop Test Plan	5 days	Tue 8/12/14	Sat 8/16/14	27,25	Testers
Integrate Testing	2 days	Mon 8/18/14	Tue 8/19/14	25,31	Testers
Integration Testing	**12 days**	**Wed 8/20/14**	**Wed 9/3/14**		
Test module integration	5 days	Wed 8/20/14	Tue 8/26/14	32	Testers
Identify anomalies to specifications	2 days	Wed 8/27/14	Thu 8/28/14	34	Testers
Modify code	3 days	Fri 8/29/14	Mon 9/1/14	35	Testers
Re-test modified code	2 days	Tue 9/2/14	Wed 9/3/14	36	Testers
Integration testing complete	0 days	Wed 9/3/14	Wed 9/3/14	37	
Final Software Documentation	**2 days**	**Thu 9/4/14**	**Fri 9/5/14**		
Review all documentation	2 days	Thu 9/4/14	Fri 9/5/14		Project Team
Documentation complete	0 days	Fri 9/5/14	Fri 9/5/14		Project Team
Deployment	**2 days**	**Mon 9/8/14**	**Tue 9/9/14**		
Test Integration	1 day	Tue 9/9/14	Tue 9/9/14		Deployment Team
Deploy software	1 day	Wed 9/10/14	Wed 9/10/14	43	Deployment Team
Post Implementation Review	**3 days**	**Thu 9/11/14**	**Mon 9/15/14**		
Software development template complete	0 days	Sat 9/20/14	Sat 9/20/14		Deployment Team

6.4.2. Schedule allocation

The project is constrained to 13 weeks only, which start on June 28, 2014 and with final deliverables due on the 3rd week of September year 2014. Schedule will be controlled with the use of Gantt chart.

5.2.3. Resource allocation

All team members will work separately on their assigned tasks. Each member is expected to devote at least 15 hours per week in the whole duration of this project. They must also be open to any changes of schedule, emergency meetings, and overtime to meet the deadline of each deliverables.

Team leader will monitor the daily progress of each member, responsible for overall quality of the project and must ensure that schedules are met, and also is responsible for the risk management.

Each member must communicate at the end of the day to report any progress and/or impediments. They must also be responsible to initiate research for any needed information.

The team must minimize faults and maximize user-friendliness in developing the system. Everyone has overall responsibility for the documentation and has to ensure that it is updated.

5.2.4. Budget allocation

5.3. Control plan

5.3.1. Requirements control plan

5.3.2. Schedule control plan

The team leader will maintain the schedule in a project document. She will be responsible for gathering the individual tasks for each team member and making the status report. If schedule is not followed because of some impediments, the team will have a schedule control meeting.

5.3.3. Budget control plan

All resources needed in development are readily available with each team member; therefore, the team has no budget control plan for this project.

5.3.4. Quality control plan

The team tester will be responsible in generating a Quality Control Plan. Gathered information from weekly meeting and reviews at the end of each phase will be used to control the quality of the work process and the resulting work.

5.3.5. Reporting plan

General Reporting

The team will make use of internet (e-mail) and mobile phones as means to communicate to each member for updates, status on task, schedules, and impediments of the project.

Internal Reporting

Each member will inform the team on the current status of tasks assigned and impediments every meeting. The team leader will gather all information and will make an analysis for a revision.

External Reporting

They will also report to the project manager every deadline of deliverables.

5.3.6. Metrics collection plan

The team will have a weekly meeting. Weekly meeting's agenda will be each member's progress report.

Each member will note their tasks assigned, accomplishments and impediments; And as well as their number of hours planned for the next week, the actual hours each has served and future hours needed to be rendered to beat the deadline. The team leader will consolidate the data and will analyze the efforts spent per members every week.

5.3.7 Risk management plan

The team identifies some risk factors that users and developers may encounter when using the system. Communication between the user and the system; without a reliable network, user may not be able to use the system. Software update may also affect the system. Some components in the system may not be functional when it is not designed with the latest technology. The more data stored in the database the more slower the system works. Team members will identify mitigation plans for all identified risks and soon to be identified risks as the project progresses

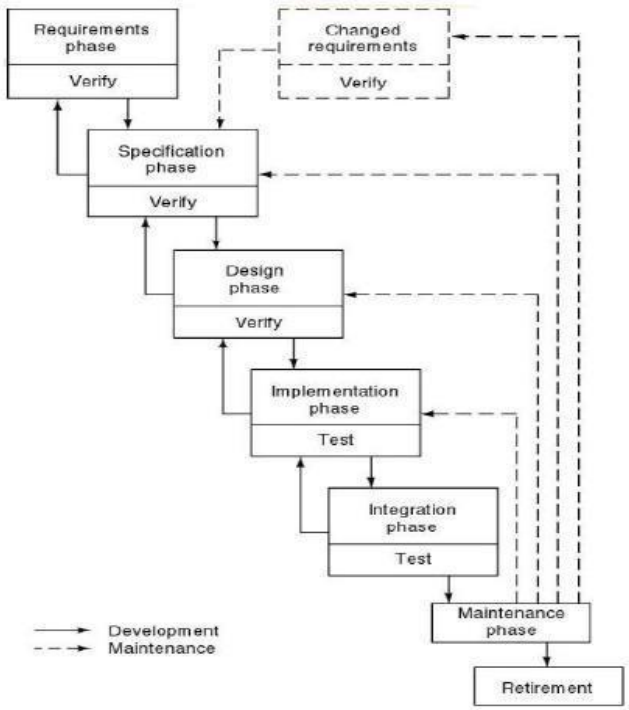
5.3.8 Project closeout plan

The project team will ensure to beat the deadline and to have proper closeout of the project on 1st week of October.

6. Technical process plans

6.1 Process Model

The team will make us of the Waterfall Model, also referred to as a linear-sequential life cycle model. The model is very simple to understand and use. Each phase has a corresponding deliverable to meet.



6.2 Methods, tools, and techniques

The methods and techniques listed in this table will be evaluated and applied in specific areas of the project as appropriate:

Category	Methods and Techniques
Requirements Elicitation	Meetings Interviews Brainstorming Consultation
Formal Specification and Analysis	Use cases to define requirements UI drafts

Methods and Techniques

Category	Tools
Operating System	Windows XP, Vista or Windows, 7, Windows 8, Mac OS
Development languages and databases	Databases: MySQL Language: PHP, HTML, CSS, Javascript
Planning and Tracking	MS Project 2007

Framework	Codeigniter, bootstrap
Software	Sublime Text, Xampp
Documentation	All document will be written using Microsoft Word, and PowerPoint / Prezi for Project Presentation

6.3 Infrastructure Plan

All members will work at their own location where own facilities needed for the development are present. Each member will communicate for any impediments and updates of the assigned task given. Compilation of complete tasks for integration will be submitted to the team lead for checking with the use electronic mail to send necessary files for the development and documentation.

6.3 Product Acceptance Plan

The project manager will evaluate each deliverable of the project. Any errors, comments and corrections will be applied for revisions. Acceptance of deliverable will only be achieved through the project manager's approval for final documentation.

Software acceptance will only be verified according to the users' response when testing the system.

7. Supporting process plans

7.1. Configuration management plan

The project will be based on approved deliverables made. Continuous checking of any defect and change of requirements will be done and documented to insure a complete working project; changes will reflect on revisions of each deliverables made.

7.2. Verification and validation plan

As the project will proceed with the implementation phase, verification and validation plan will be done by the team tester. Continuous testing of different parts of the system will be done and properly documented to insure that each functions stated on the SRS are implemented.

A regular testing by co-members will also be done to review progress of work done.

The following are the types of testing that the group will be utilizing:

Black Box Testing

Testing method wherein it relies on testing the software with various inouts and validating results against expected.

Boundary Value testing

This technique is done extensively to check for defects t boundary conditions.

Big Bang Integration Testing

This will be done together with other subsystems that the project will be linking.

Browser Compatibility Testing

It is performed to check the compatibility of the project to different browsers commonly used.

Condition Coverage Testing

This is to test all condition statements in the code.

Functional Testing

This will focus on testing the software against the SRS and SDD.

GUI Testing

This is to test the software's GUI if it meets the requirements as mentioned in the SDD.

Acceptance Test

It is a formal testing performed by the client to check if the software has met their requirements.

7.3. Documentation plan

Each deliverables has corresponding documents. All team members are responsible in producing the documents. The lists of documents that will be created and maintained under version control include:

Software Requirements Specification (SRS)

Software Project Management Plan (SPMP)

Software Design Description (SDD)

Software Testing Document (STD)

Test Reports

Note: Not all listed here are to be submitted/delivered to the client.

7.4. Quality assurance plan

The quality assurance plan will done together with the Verification and Validation Plan (refer on section

7.2).

7.5. Reviews and audits

The review and audits will depend on the Verification and Validation Plan, since there will be various testing types to be done for this project.

7.6. Problem resolution plan

Each member is expected to allocate extra hours for handling any impediments of any project phases. Backlogs and preliminary bug fixing are already considered in the work breakdown structure, especially during the implementation phase.

7.7. Subcontractor management plan

Though the team is only composed of 4 members, subcontractors for this project will not be needed.

7.8. Process improvement plan

Meetings and other means of communication will be done to evaluate progress of task and efficiency of the team members. This will see the improvements of the project and will be properly documented for the process improvement plan.

7.9 Additional Plans

9. Plan Annexes

10. Index

Software Design Description

for

CIT-U Events Calendar

Signature

Members	***Role***	***Signatures***
Berbo,Chris Ian	Team Leader/Tester	
Ragasajo, Jan Michael	Developer/UI Designer	
Judan, John Mike	Developer/Analyst	
Bacarisas, Joshua	Developer/Tester	
Lecciones, Patrick	Documentation/ Tester	

Change History

Name	Date	Changes
Preliminary Version	August 16, 2014	**Basic design of the website is changed**
Secondary Version	September 18, 2014	**Mobile version is removed, new UI is added, Super admin functions are added and holiday events added.**
Tertiary Version	September 27, 2014	**Date range function added, dummy creation removed, new modal added, Recolors done to the main page**

Preface

This is the software design description (SDD) document for the CIT-U Events Calendar. In particular, this document will provide the design details for this system and will address the work made by our group in proficient English language explaining the functions of the system.

The purpose of this project is to assemble under one cover a sufficient body of knowledge about managing a successful software project. This SDD is intended to be used by the CIT-U executive committee for the purpose of evaluating the groups' contribution on the project for implementing the design of the said project.

Table of Contents

List of Figures

Communication Diagram

Entity Relationship Diagram

List of Tables

7. Introduction

7.1. Purpose

Our main purpose of this documents is to provide basic UI and functionalities for our web project and mobile system.

1.2 Scope

Scope for this documentation is for the client and our project manager.

1.3 Definitions and Acronyms

Acronym/Words	Definition
IEEE	Institute of Electrical and Electronics Engineers
SRS	Software Requirements Specifications

2. References

- IEEE 12207.2-1997 Industry Implementation of International Standard ISO/IEC 12207: 1995 (ISO/IEC 12207)Standard for Information Technology - Software Life Cycle Processes - Implementation Considerations
- IEEE Std. 830-1998 IEEE Recommended Practice for Software Requirements Specifications
- [IEEE] The applicable IEEE standards are published in “IEEE Standards Collection,” 2001 edition.
-
- www.google.com

3. Decomposition Description

3.1Module Decomposition

We first look at the modular decomposition of the software. Next, we look at the concurrent process description of the software. Lastly, we will look at the different execution threads which are possible.

3.1.1Module 1 Description

This module discusses about the game. This module also describes the mechanics of the game, how to play the game and the admin part to make the game dynamic.

3.1.2 Administrator Module Description

The user subsystem has a GUI that allows the user to choose which month he or she wants to display, then this provides information about that month like events, memos, announcement, etc..

3.1.3 User Module Description

Identification	User(web)
Type	Module
Purpose	To allows the user to choose what month to display information or details about the future or current happenings in the chosen month.

Table 3.11 – User/Client Module Description

3.1.3 Admin Module Description

Identification	Admin(web)
Type	Module
Purpose	To allow the admin to add announcements to a specific month that will be later on displayed to the user side.

Table 3.1.2 – User/Client Module Description

3.1.3 Super Admin Module Description

The costumer subsystem has a GUI that takes the user directly to the selection menu.

Identification	Super Admin(web)
Type	Module
Purpose	To allow the admin to post events, announcements and memo's . The super admin can block, approve and move to what department the admin currently belongs.

Table 3.1.3 – Costumer(web) Module Description

3.2Concurrent Process Decomposition

3.2.1 Database Process Description

Identification:	Database Process	
Type	:	PHP file
Purpose	:	To access the database and perform all the queries.
Function	:	Waits for query request in managing entities.
Lifetime	:	The duration of the system's life.
Subordinates:		Apache Web Server and MySQL Server.

3.2.2 Server Process Description

Identification:	Server Process	
Type	:	PHP file
Purpose	:	To manipulate the record of participants and events.
Function	:	Select Operation to perform (e.g.: add, edit, delete) Prepare and display interface on requested command. Perform queries according to requested command.
Lifetime	:	The duration of the system's life.
Subordinates:		Apache Web Server and MySQL Server.

3.2. 3 Web Process Description

Identification	:	View content
Type	:	HTML and CSS File
Purpose	:	To help the client/user view contents in the website.
Function	:	Lets the user select a month to display its contents.
Lifetime	:	The duration of the system's life.
Subordinates:		Apache Web Server and MySQL Server.

4. Annexes

4.1. Data flow diagram (optional)

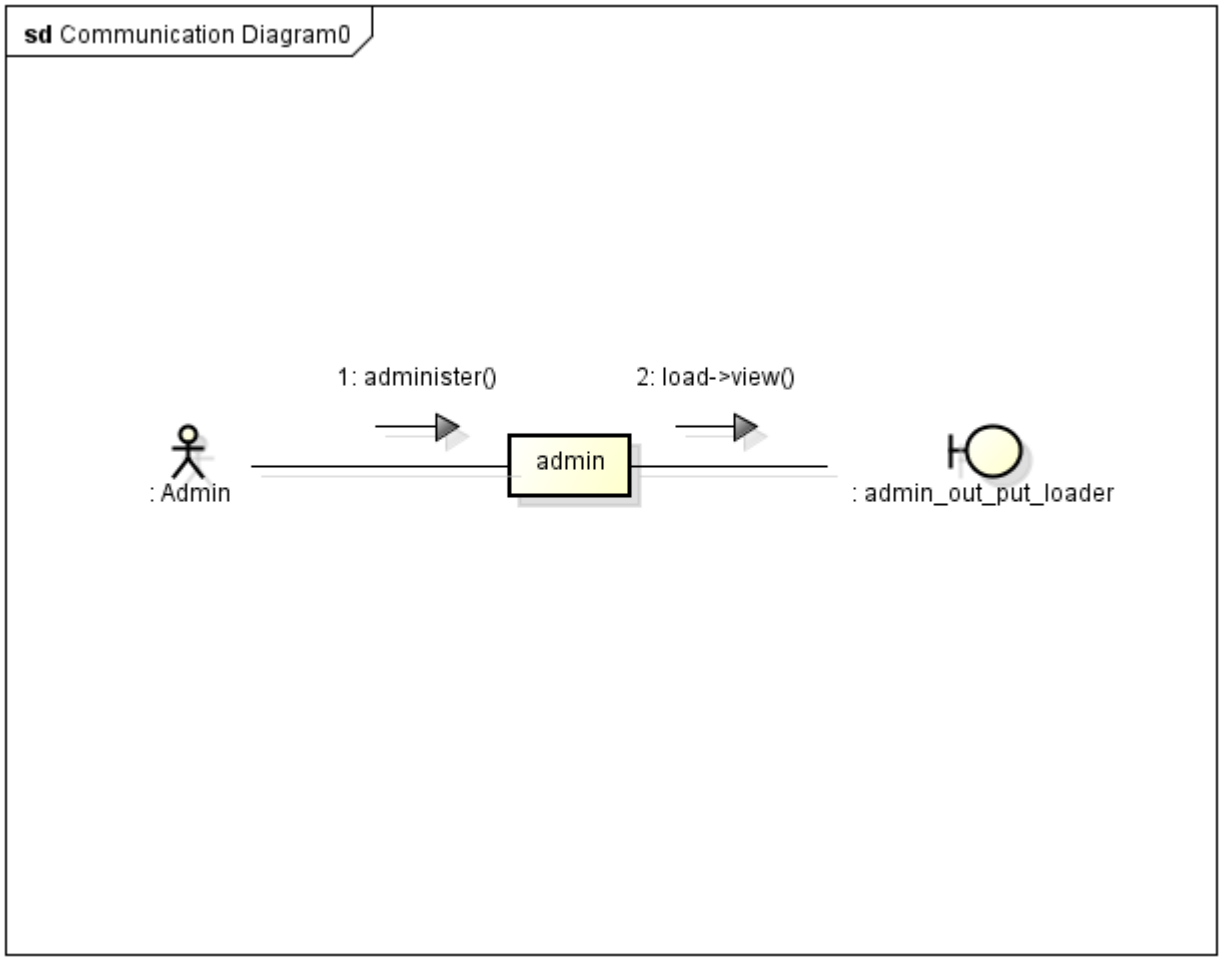
None

4.2. Class diagram

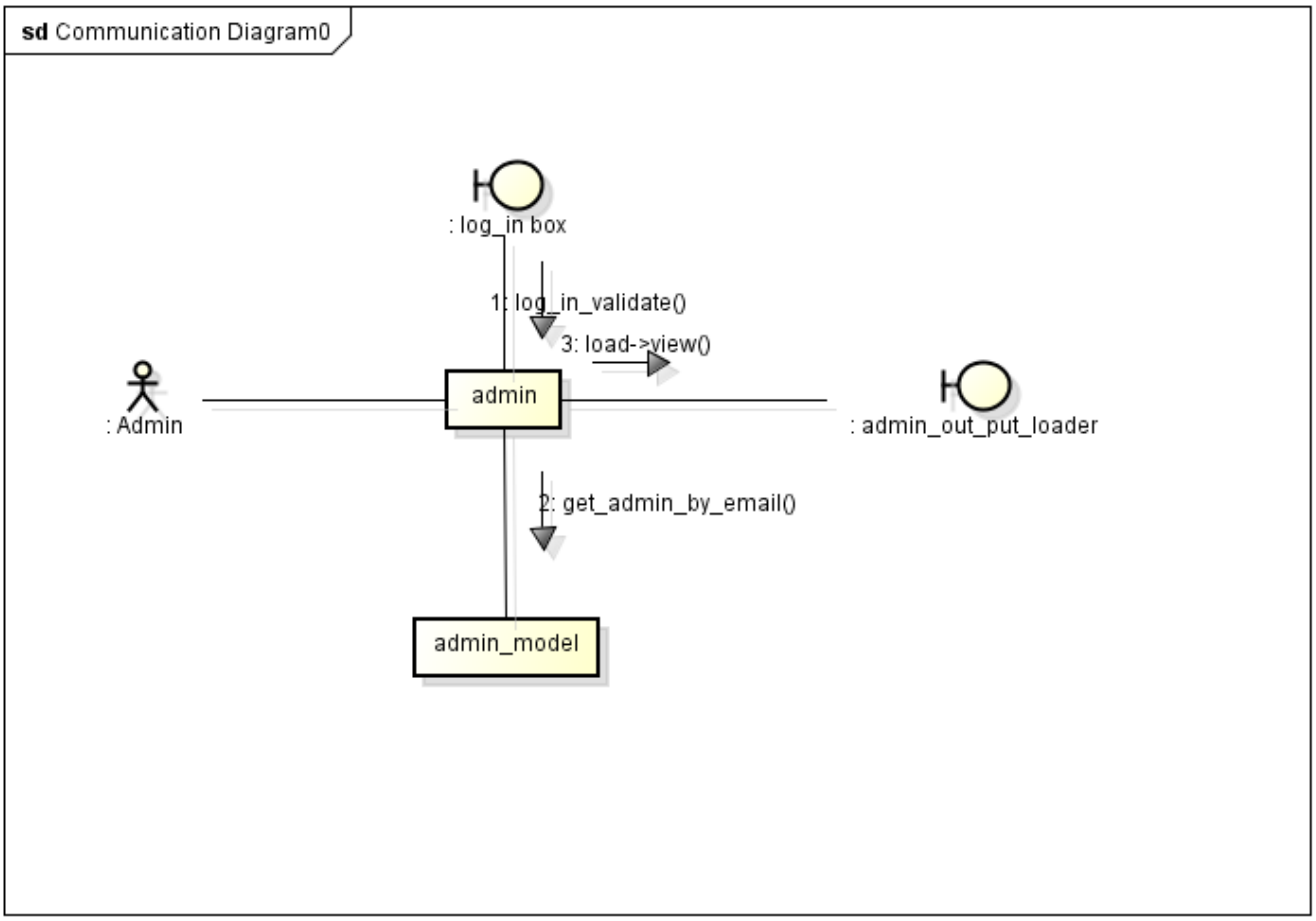
None

4.3. Use case realization (Sequence diagram / Communication diagram)

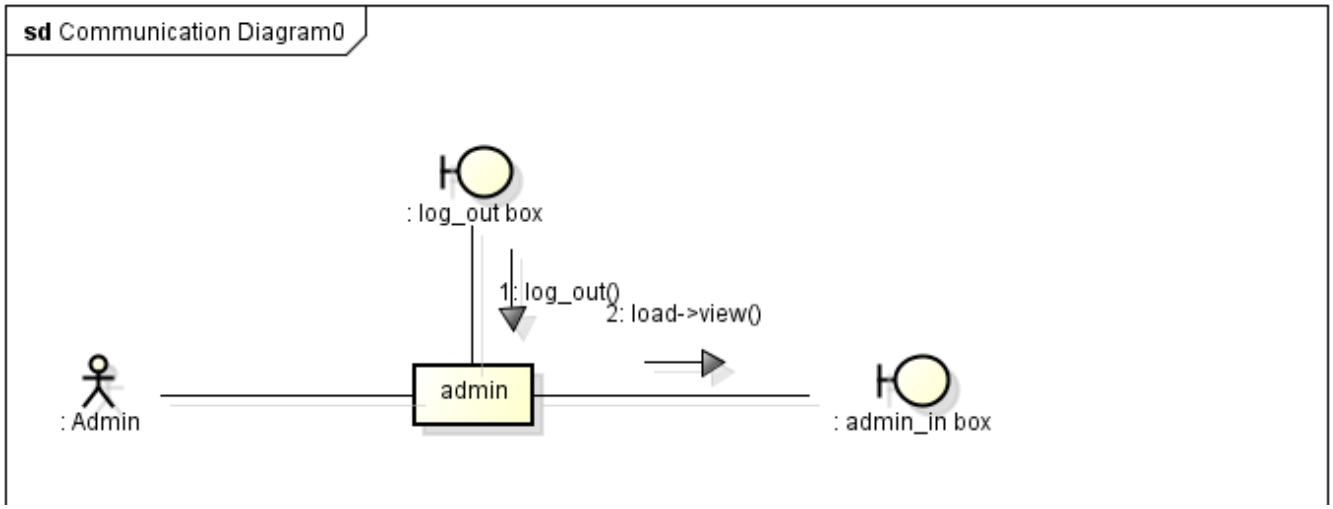
Admin-Administer



Admin-Login

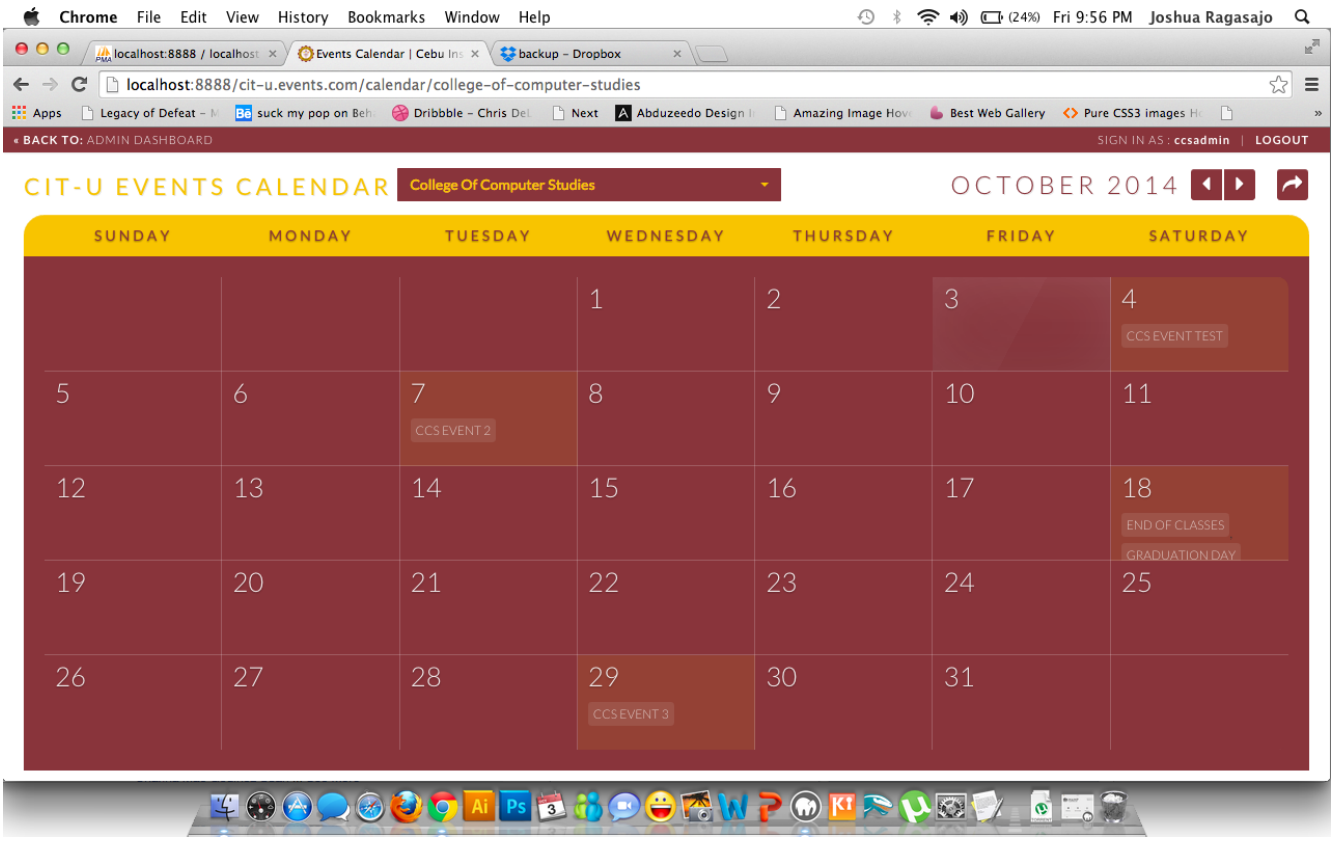


Admin-Log out

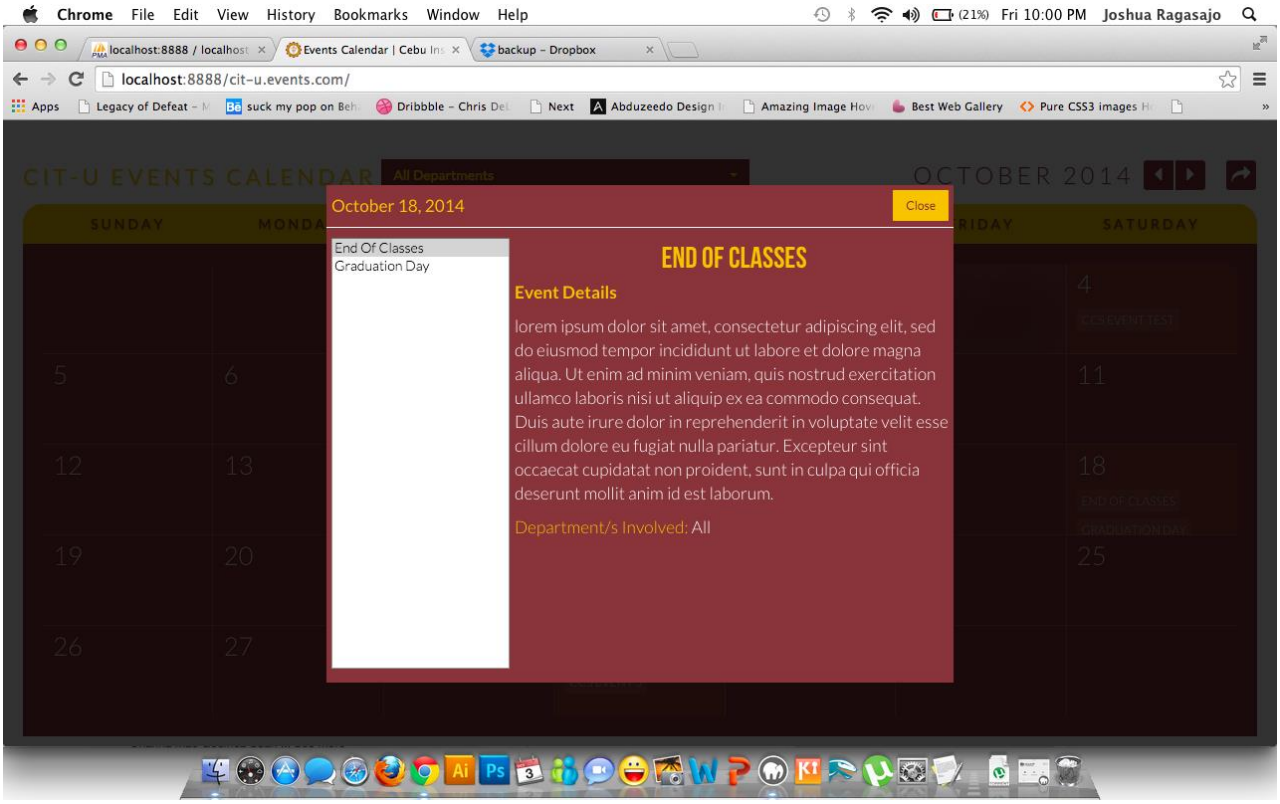


4.4. User interface design

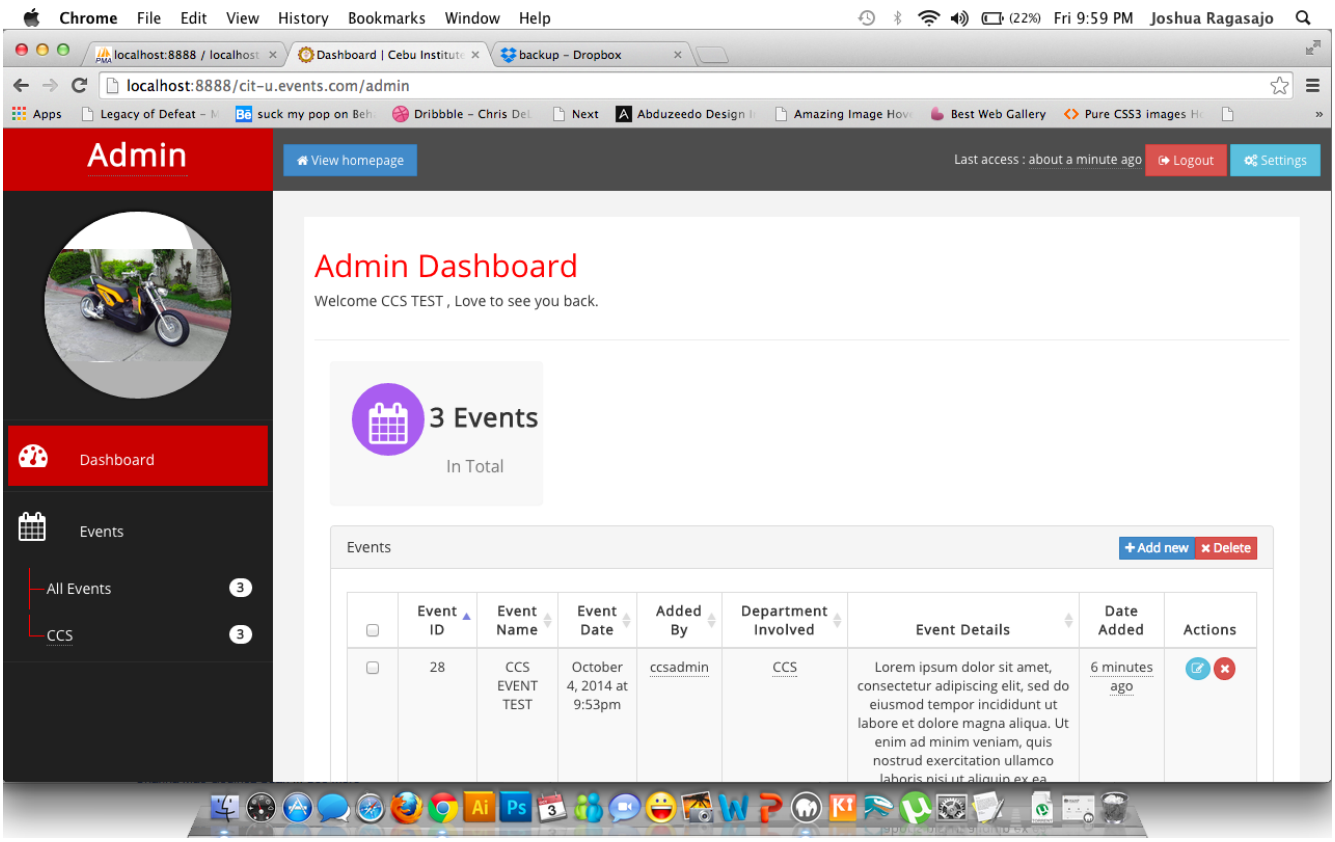
Main Page



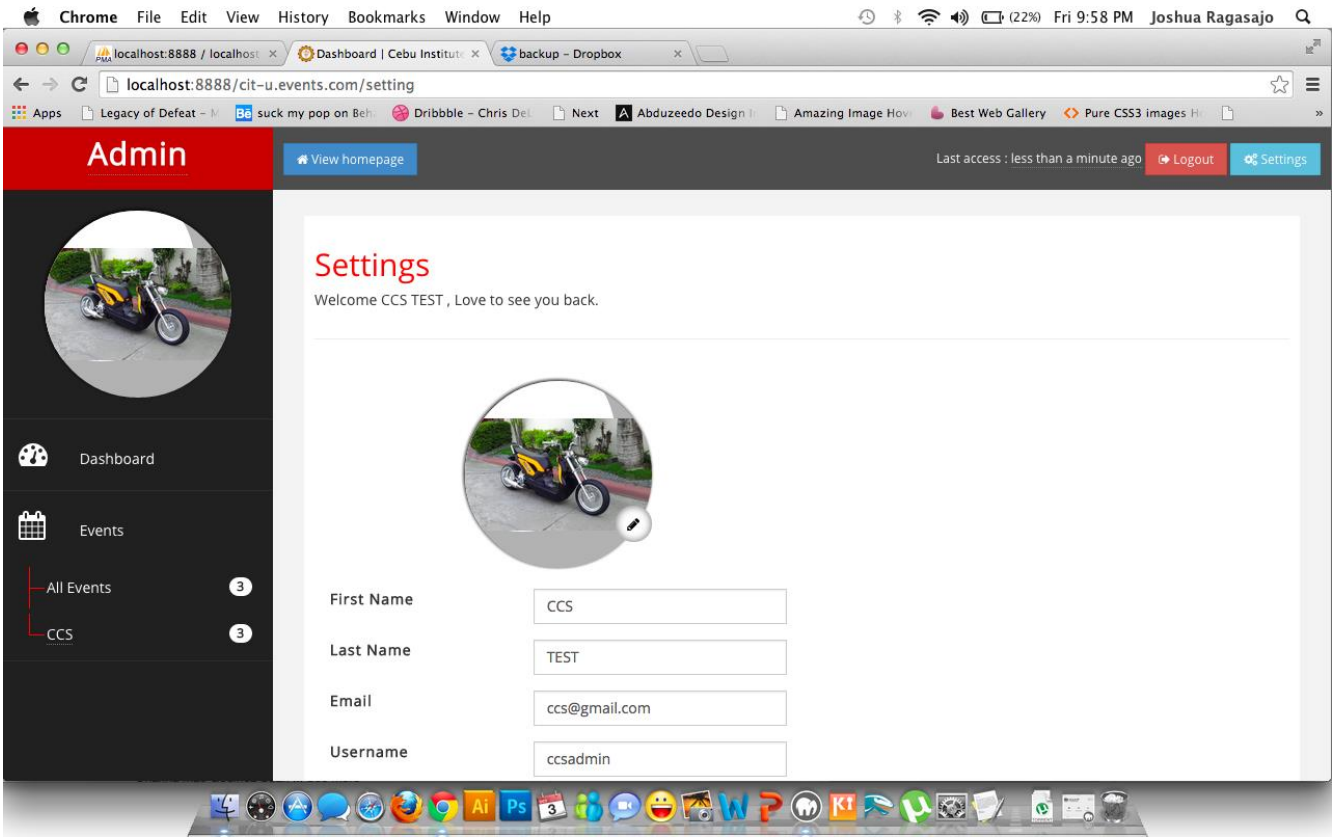
Events Modal



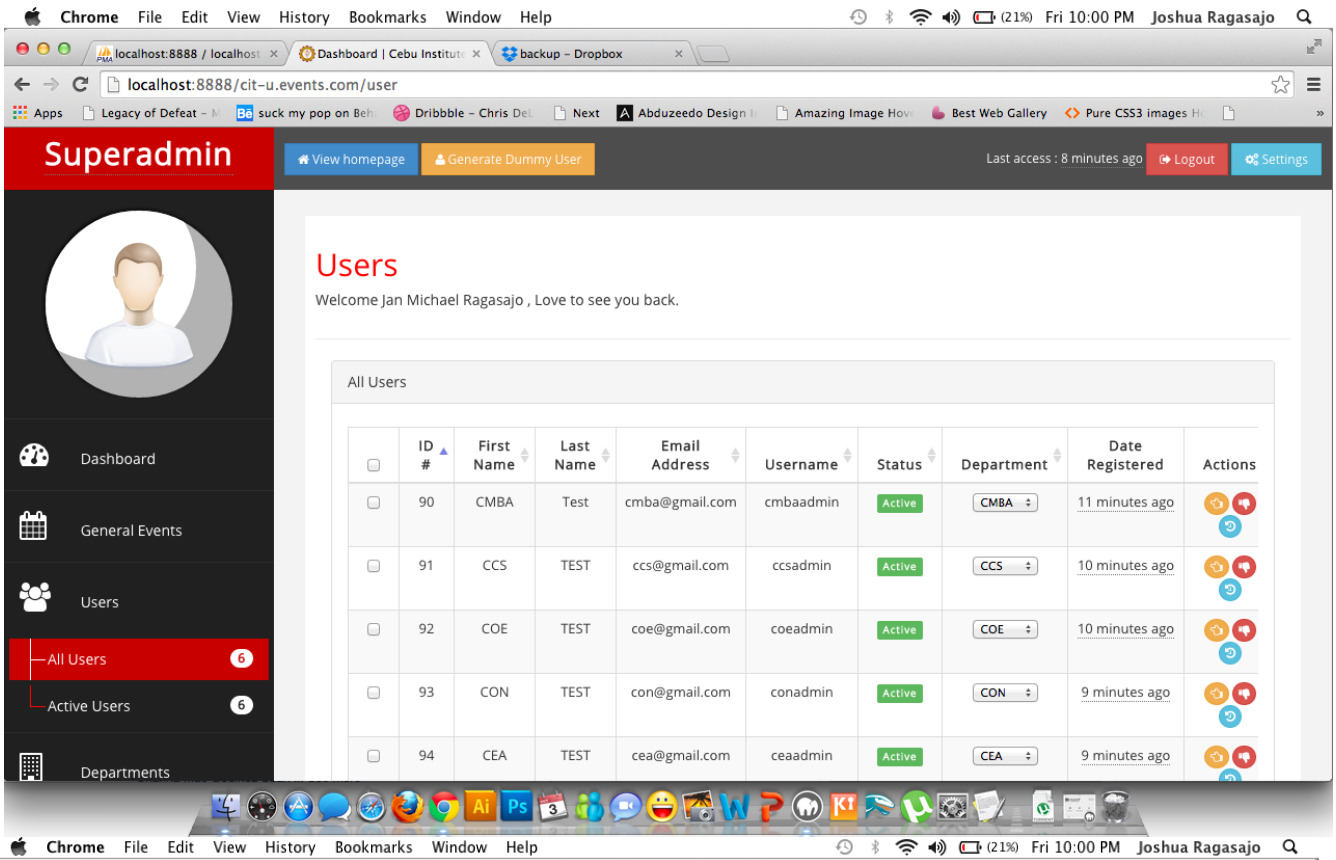
Admin Dashboard

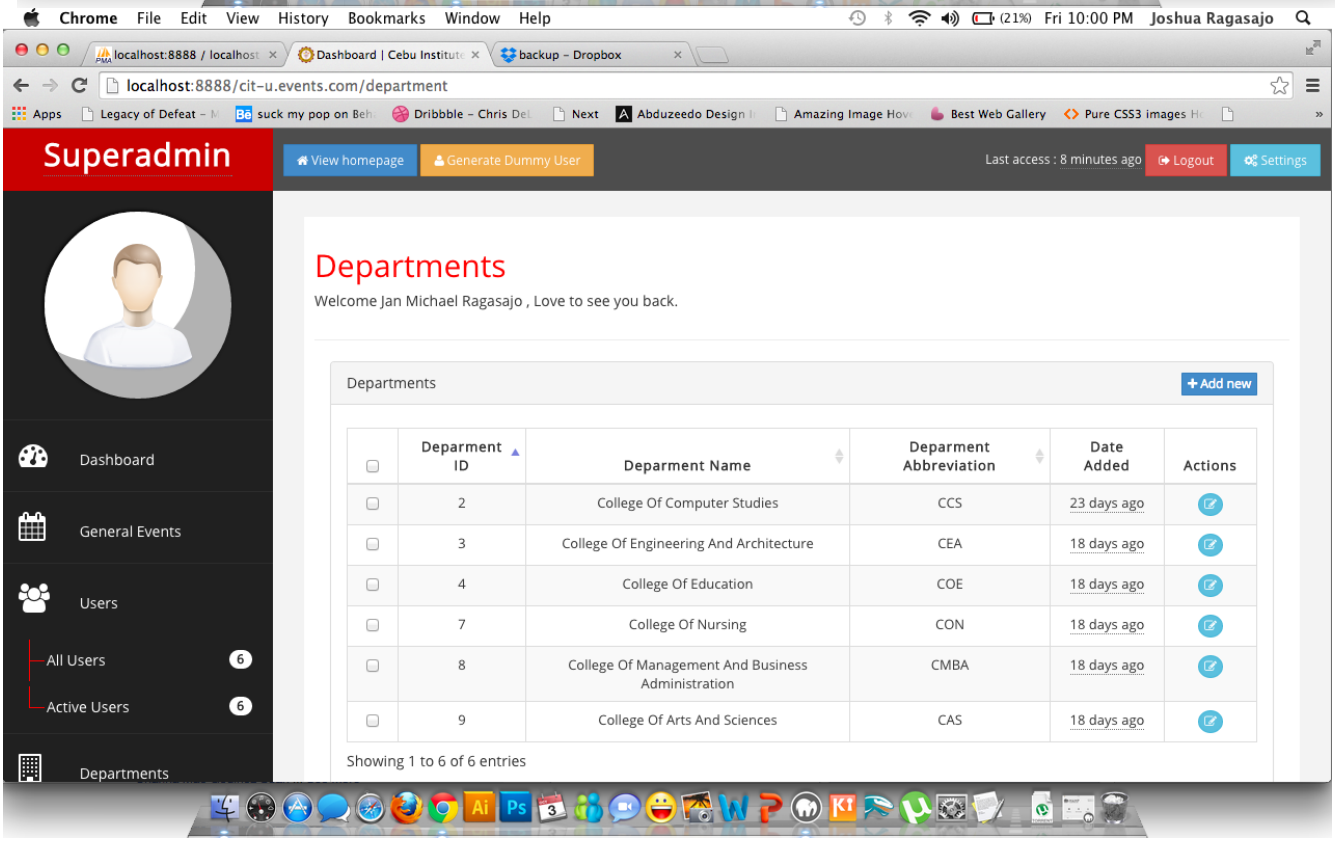


Admin Profile Settings

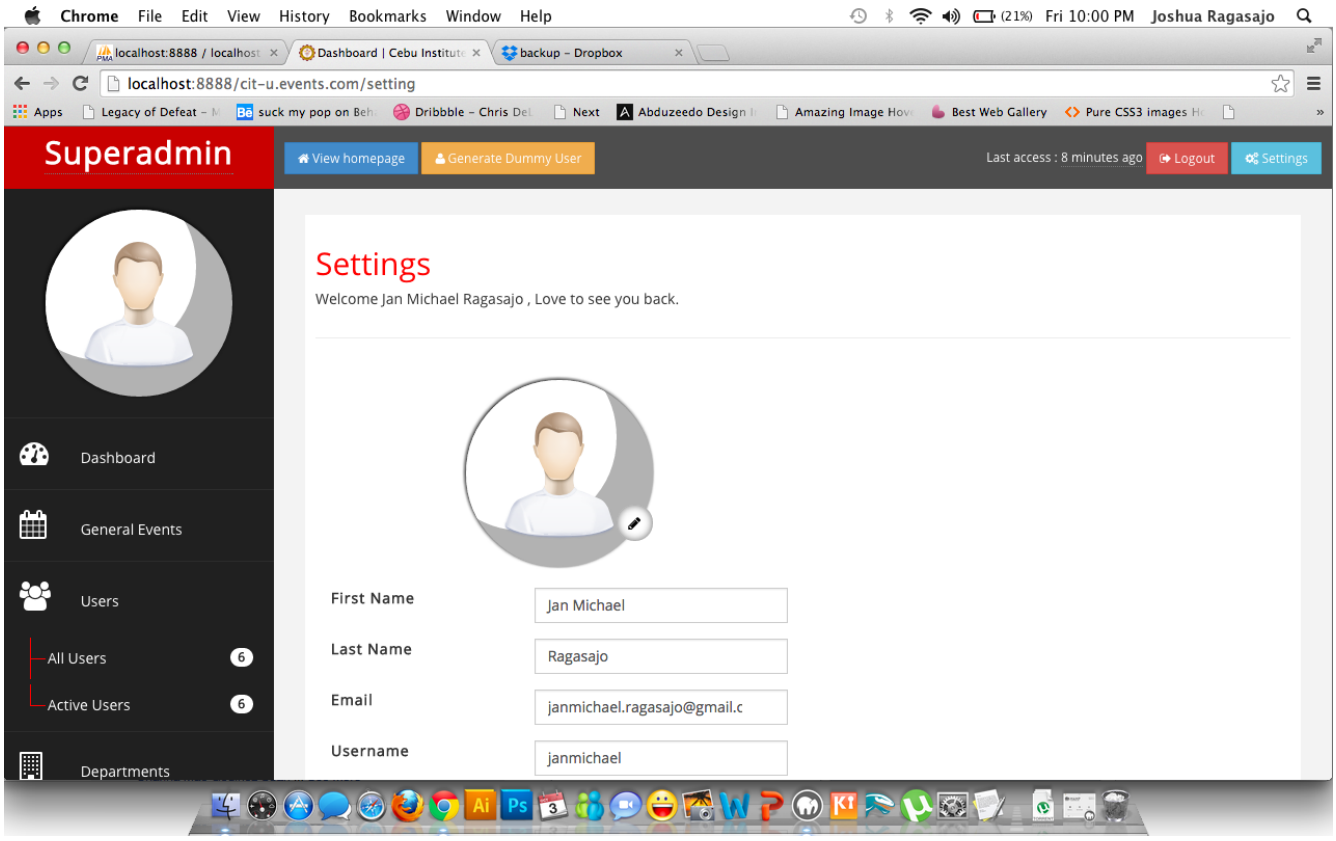


Super Admin Dashboard-Functions





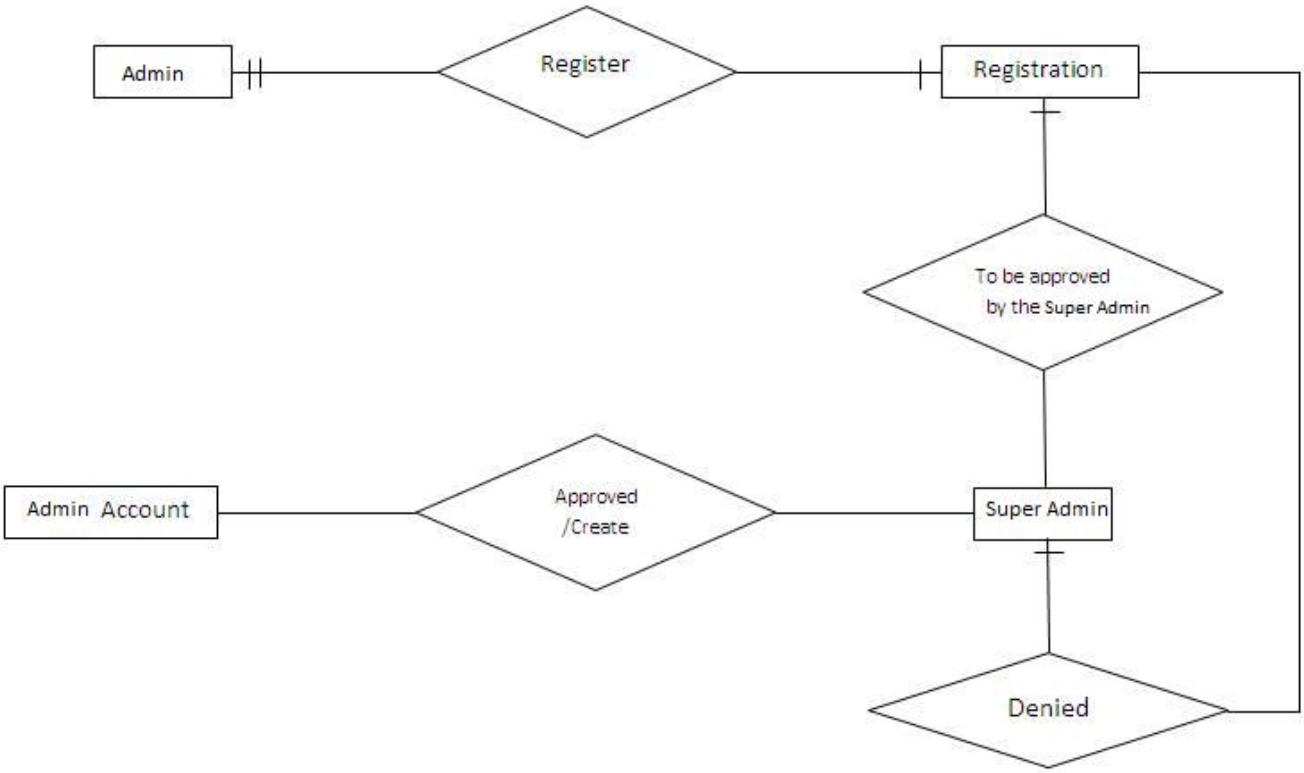
Super Admin Profile Settings



4.5. Entity-relationship diagram

Admin – Super Admin ERD

Registration ERD



Software Test Documentation

for

CIT-U Events

Signature

Members	Role	Signatures
Berbo,Chris Ian	Team Leader/Tester	
Ragasajo, Jan Michael	Developer / UI Designer	
Judan, John Mike	Developer / Analyst	
Lecciones, Patrick	Documentation/Tester	
Bacarisas, Joshua	Developer / Tester	

Change History

Name	Date	Reason For Changes
Preliminary Version	August 16, 2014	**Basic design of the website is changed**
Secondary Version	September 18, 2014	**Mobile version is removed, new UI is added, Super admin functions are added and holiday events added.**
Tertiary Version	September 27, 2014	**Date range function added, dummy creation removed, new modal added, Recolors done to the main page**

Preface

This Software Test Documentation is patterned from the: IEEE Std. 829-1998 IEEE Standard for Software Test Documentation; from which, selected and applicable sections were taken, and modifications to certain headers were made in order to curtail the needs of this project.

This document serves as the official structure of the testing phase of Name It. A substantial amount of information herewith is taken from the SRS and SDD but is not discussed in detail; except for the ones related to testing.

Table of Contents

8. Scope

This document is designed in such a way that all the features, parts, and details concerning CIT-U Events, are detailed here. Guidelines as to how each of the classes and features will be tested will be discussed in their respective section.

This document gives heavy emphasis and exhausts all functionalities from application loading, to the level templates, options, and the exit command. Each feature is expounded in such a way that all its quantifiable components are listed and shown with respect to how they are tested and evaluated, and the actions that will be done should any of them not function as intended.

9. References

IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology.

Karl E. Wiegers SOFTWARE REQUIREMENTS, 2003

Name It Software Requirements Specification

Name It Software Design Description

10. Definitions

Term	Definition
SRS	Software Requirements Specification
SDD	Software Design Description
Design level	Software Design decomposition
Software feature	A distinguishing characteristic of a software item
Software item	Any substantial code/s within the program

11. Test Plan

11.1. Purpose

The Test Plan will showcase the entirety of the testing procedure. It aims to give the reader an idea as to how the test will be conducted, what features will and will not be tested, and the planned actions based on the given outcome.

11.2. Outline

11.2.1. Test Plan Identifiers

This document will use the following codes as identifiers for their respective sections:

- NI_1 – TP [the Test Plan]
- NI_2 – DS [the Design Specification]
- NI_3 – CS [the Case Specification]
- NI_4 – TD [the Test Documentation]
- NI_5 – TL [the Test Log]
- NI_6 – IR [the Incident Report]
- NI_7 – TS [the Test Summary]

11.2.2. Introduction

This test plan explores every part of Name It as it was outline in the SRS. It will use every use-case diagram as a guide to how each test instance will be treated. The timestamp recording of each test will be taken into account but will not weigh in any value to the overall result of each test.

The test plan will include the test procedure, expected outcome, pass/fail criteria, but will not include maximum execution times as this varies greatly from each Android phone's processing capability. Items to be tested

The following items (detailed in the SRS and SDD) will be tested:

- Landing page
    - Calendar
        - Select Department
        - Select Month
        - Select/View Event

11.2.3. Features to be tested

Main page: This feature involves the UI [MainMenuView] and control [MainMenuController] of the main screen that has 2 subfunctions namely:

- Select Course [gives the user the freedom to choose which school event/department they want to view]
- Select Month [which gives the users list of months and the user can select which month the user wants to view]
- Select Day(with mark/event)[lets the user view the specific details about that day.]

11.2.4. Approach

The testing approach used is a modified top-down approach. Simply, this is going through every possible combination of UI commands shown in each of the respective UI's as detailed in the SRS and SDD. The test will be done this way:

1. Testers will start the application and go through the project on a linear approach without pressing the back/return button. At the end of each routine, another combination will be tried.
2. Testers will now attempt to press the back/return button every time a feature is selected.

11.2.5. Item Pass-Fail Criteria

Each feature must perform exactly as the SRS and SDD describe them to be. Although there is no minimum time response for each feature, none of them should take 5 seconds to load, at most.

11.2.6. Suspension criteria and resumption requirements

Should the team encounter a deadlock at any point in the testing phase, it will be noted, and 2 succeeding tests will be issued to measure the consistency of the failure/bug. If it is prevalent, then the testing will be suspended.

Resumption of testing will occur when the bug has been found and the code has been restructured in such a way that the latter will not occur again.

Every test is repeated at least 3 times to ensure an accurate and dependable execution.

11.2.7. Test deliverables

The following documents will be included here:

- STD [CIT-U Events]
- Test Design Specifications
- Test Case Specifications
- Test Logs
- Test Incident Report Logs
- Test Summary Reports

11.2.8. Testing tasks

- Operation Testing
- Re – Test

12. Index