**[Type the company name]**

[Year]

**[Type the document title]**

**[Type the document subtitle]**

**ng**

**Table of Contents**

**Group Introduction**

**Group number:** 1

**Number of members in group:** 5

**Member names, Student ID and assigned roles:**

1. Ng Shien Wee; 4519589 (Team Manager)
2. Damon Roy Walker; 4277247 (Lead tester)
3. Giritharan Govindasamy; 3934767 (Lead Designer)
4. SiuTung Lee, 3775896 (Lead programmer)
5. Lee Toby Yin Kiu, 4250941 (Lead documenter)

\*Assigned roles to a person do not mean that they are fixed to their role. They are flexible to work with other group mates on different tasks.

**Artefacts delivered by members:**

|  |  |  |
| --- | --- | --- |
| **Artefacts** | **Submitted by** | **Submitted on** |
| Business Case | Lee Toby Yin Kiu | 25/8/2014 |
| Detailed plans | Ng Shien Wee | 10/9/2014 |
| Risks and counter measures | Lee Toby Yin Kiu and Damon Roy Walker | 27/8/2014 |
| Use cases | SiuTung Lee and Damon Roy Walker | 5/9/2014 |
| Use case description | Ng Shien Wee | 7/9/2014 |
| Domain model | Lee Toby Yin Kiu and Giritharan Govindasamy | 5/9/2014 |
| Data dictionary | Ng Shien Wee | 7/9/2014 |
| Software requirement specification | Ng Shien Wee, Damon Roy Walker and Lee Toby Yin Kiu | 6/9/2014 |

**Member Contribution Assessment:**

|  |  |  |
| --- | --- | --- |
| **Name** | **Contribution** | **Signature** |
| Ng Shien Wee | Contributed |  |
| Lee Toby Yin Kiu | Contributed |  |
| Damon Roy Walker | Contributed |  |
| Giritharan Govindasamy | Contributed |  |
| SiuTung Lee | Contributed |  |

**Business Case**

**System Request –** Conference Management System

**Business Need:** This software system is being developed to help conference organizers to cope with the complexity of the refereeing process.

**Business Requirements:**

Using this system, the process of organizing a conference would be simplified. By using the system, we can expect some processes, which are currently conducted manually, to be performed automatically or semi-automatically by the system. The conference management system should have the following key features:

* Management and monitoring of the program committee
* Sophisticated and flexible management of the access of PC members and referees to paper and conflicts of interests
* Paper submission
* Automatic and manual paper assignment based on the preferences of PC members
* List of the latest events
* Submission of reviews
* Automatic preparation of conference proceedings
* Discussion of reviewing of papers
* The author response (aka rebuttal) phase, when the author can respond to the reviews
* Sending email to PC members, referees and authors (optional)
* Monitoring email (optional)

**Business Value:**

We can expect that there will be more authors who are willing to take part in the conference and be able to go through the process of submission of paper easily. The system can help promote the conference by being easy to use, reliable and always available. The software system also aims to minimize the number of people involved in the technical aspects of the conference set up and reduces the cost of organizing a conference. The system would store data digitally to reduce the amount of paper work.

**Special Issues or Constraints:**

* The organization would like us to use C++ Programming language for the system and expect the system to be easy to use, highly secure, reliable and available.
* The system should be ready by the end of October.

**Detailed Plans**

**Gantt Chart**

**Activity Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **Activities** | **Expected Due** | **Responsible** | **Comment** |
| **Planning Phase** |  |  |  |
| Initial project planning | 11/8/2014 | Wee, Damon, Toby, Girit (**Completed**) | Have a good understanding of the system that we are building and what artefacts are to be submitted |
| Online repository | 1/9/2014 | Wee (**Completed**) | Have a cloud repository to share the files |
| Detailed planning | 10/9/2014 | Wee (**Completed**) | Coming up with detailed plans |
| **Requirements Elicitation** |  |  |  |
| Question formation | 12/8/2014 | Wee, Damon, Girit, Toby (**Completed**) | Everyone comes up with questions and combined the questions together on the due date |
| Interviewing of clients | 13/8/2014 | Everyone (**Completed**) | Met with client for the first time and get answers for the first list of questions |
| Question refining | 20/8/2014 | Wee (**Completed**) | Refine the questions that we have prepared and some answered to obtain more detailed requirements from client |
| Interviewing of clients 2 | 21/8/2014 | Everyone (**Completed**) | Get detailed requirements from client |
| **Documentation (includes editing)** |  |  |  |
| Business case | 25/8/2014 | Toby and edited by Wee (**Completed**) | The business needs, requirements, value and constraints |
| Risks and counter measures | 27/8/2014 | Toby and Damon and edited by Wee (**Completed**) | Coming up with the risks and counter measures |
| *Software Requirements Spec* |  |  |  |
| Introduction | 1/9/2014 | Toby and edited by Wee (**Completed**) |  |
| Overview | 8/9/2014 | Manfred and edited by Wee (**Completed**) |  |
| Functional Requierement | 3/9/2014 | Wee (**Completed**) |  |
| Non-functional Requirement | 3/9/2014 | Girit and edited by Wee (**Completed**) |  |
| Use case model | 5/9/2014 | Damon, Wee and Manfred (**Completed**) |  |
| Use case description | 8/9/2014 | Wee (**Completed**) |  |
| Class/Domain model | 5/9/2014 | Girit, Wee and Toby (**Completed**) |  |
| Data dictionary | 8/9/2014 | Wee (**Completed**) |  |
| Compilation | 11/9/2014 | Wee (**Completed as of 11/9/2014**) |  |
| **Implementation (Coding)** |  |  |  |
| Designing | Week 12 | Girit (**Incomplete)** | Designing the command line program |
| Use case 1-5 | Week 8 | Wee, Manfred and Girit (**Incomplete**) | 5 use cases done per week till final submission |
| Use case 6-10 | Week 9 | Wee, Toby and Damon  (**Incomplete**) | 5 use cases done per week till final submission |
| Use case 11-15 | Week 10 | Manfred, Girit, Toby (**Incomplete**) | 5 use cases done per week till final submission |
| Use case 16-23 | Week 11 | Toby, Damon, Wee (**Incomplete**) | More use cases done to provide leeway for integration of codes |
| Use case 24-27 | Week 12 | Wee, Girit (**Incomplete**) | 3 use cases this week |
| Integration of codes | Week 12 | Wee (**Incomplete**) | Integration of codes and submission |

**Risks**

The tables below will give detailed explanations about the risks that we may encounter when working as a group, which may hinder our progress and delay the project. The counter measures for the risks will also be detailed. The risks may be differentiated into different groups, including commitment, implementation and design, files, time and requirements. The severity shows how serious the risk could hinder the progress of the project. Different levels of severity include low, medium or high.

* Low: Would not hinder much of the progress of the project but should not be taken lightly
* Medium: Would hinder a significant amount of the progress of the project and should be resolved.
* High: Would hinder most of the progress of the project and should be resolved as soon as possible. More effort is required from all group members to resolve it.

**Commitment**

|  |  |
| --- | --- |
| **Risk #:** R01 | **Severity:** Medium |
| Description: The lack of commitment by the group members. This could be caused by the lack of motivation provided by the group manager towards its group members. Poor distribution of tasks by the group leader for the group members could also contribute to the lack of commitment. Furthermore, group members who do not share the same vision towards completion of task could also be a cause of the commitment issue. | |
| Counter Measure: Group members should motivate each other to complete tasks together. Proper knowledge on the strength and ability of each member is crucial to ensure proper distribution of work. A knowledgeable and experienced manager should be chosen to lead the group. | |

**Implementation and Design**

|  |  |
| --- | --- |
| **Risk #:** R02 | **Severity:** Low |
| **Description:** This is the first time where we are developing a system of such complexity. The lack of knowledge in terms of C++, designing the system and documentation may slower the process towards completion of the whole project. In terms of C++, we may encounter much more bugs and would take up more time to debug it. In terms of designing, we may lack the knowledge of domain models or use cases. In terms of documentation, we may lack the knowledge to document the requirements, non-functional requirements and much more in a structured format. | |
| **Counter Measure:** Group members should put in much more effort to do more research, do more readings and gain opinions from specialists of the field to overcome the issue. | |

**Files**

|  |  |
| --- | --- |
| **Risk #:** R03 | **Severity:** Medium |
| **Description:** There is a chance where documentations stored on the group member’s computer may be lost due to a hardware malfunction. There is also a possibility where the group might lose track on the files and updates are implemented incorrectly. | |
| **Counter Measure:** Cloud storage such as Google Drive or Dropbox could be used to store documents. To ensure proper versioning of documents, software such as TortoiseSVN or GitHub can be used. They are easy to use and highly available. We also plan to store the files and documents in multiple place just in case either one of them goes down and become inaccessible for some reason (E.g. File corruption) | |

**Time**

|  |  |
| --- | --- |
| **Risk #:** R04 | **Severity:** Medium |
| **Description:** Being full time students in university, we are always busy with assignments from other subjects. Poor time management by the group members can delay submission of artefacts. Group members may also suddenly fall sick or face unexpected problems, making them unable to accomplish the task assigned to them. | |
| **Counter Measure:** Each group member should have proper time management so that tasks assigned to them regarding the project and assignments from other subjects are completed in time. Group members who encounter an issue which may delay the submission of an artefact should update the manager as soon as possible. This would allow the task to be delegated to another member. | |

**Requirement**

|  |  |
| --- | --- |
| **Risk #:** R05 | **Severity:** High |
| **Description:** There is a possibility that during the requirement elicitation phase where the group has misunderstood what the client was looking for. The client might also decide to change the requirements during the execution of the project. | |
| **Counter Measure:** The group must constantly be in contact with the client to reconfirm the requirements and be able to obtain a clear description of them. Proper planning is essential to ensure requirement changes do not heavily impact the progress of the project. Questions to the client must also be clear and constructed. Breaking down the system into smaller parts is an example on how to overcome the issue. | |

Conference Management System

Software Requirements Specification

Version 1.5

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 01/Sep/2014 | 1.0 | First Version of SRS | Ng Shien Wee  Toby  Damon  Manfred  Girit |
| 4/Sep/2014 | 1.1 | Updated the introduction of SRS | Toby  Damon |
| 7/Sep/2014 | 1.2 | Updated requirements and overview | Ng Shien Wee |
| 10/Sep/2014 | 1.5 | Updated requirements | Ng Shien Wee |

Table of Contents

1. Introduction 4

1.1 Purpose 4

1.2 Scope 4

1.3 Definitions, Acronyms and Abbreviations 4

1.4 References 5

1.5 Overview 5

2. Overall Description 5

2.1 Product Perspective 5

2.1.1 System Interfaces 5

2.1.2 User Interfaces 5

2.1.3 Hardware Interfaces 5

2.1.4 Memory Constraints 5

2.1.5 Operations 6

2.2 Product Functions 6

2.3 User Characteristics 6

2.4 Constraints 6

2.5 Assumptions and Dependencies 7

3. Specific Requirements 7

3.1 Functional Requirements 7

3.1.1 Users Management Subsystem 7

3.1.1.1 Administrators Side 7

3.1.1.2 Program Committee Chair Side 8

3.1.1.3 User Side 9

3.1.1.4 System Side 10

3.1.2 Paper Management Subsystem 10

3.1.2.1 Paper Submission 10

3.1.2.2 Paper Assignment to Reviewers 11

3.1.2.3 Conference Proceeding 14

3.1.3 Paper Review Management Subsystem 14

3.1.3.1 Program Committee Chairs Side 14

3.1.3.2 Reviewers Side 16

3.1.3.3 Authors Side 17

3.2 Non-Functional Requirement 18

Software Requirements Specification

# Introduction

## Purpose

The purpose of this document is to describe the external behaviour of a Conference Management System. It also provides a detailed overview of the software product its parameters and goals. Functions that may be implemented later are also documented. The document also describes the non-functional requirements, and other factors necessary to provide a complete and thorough description of the requirements for the system.

This document is targeted towards the developers of the software and the potential users of the system.

## Scope

The software system to be produced is a conference management system, based on a command line interface. The conference management system aims to serve three groups of users; Program Committees (PC), PC chairpersons and paper authors. The system will be controlled by the system administrator and will be able to undertake all the groups' roles.

One of the roles of the system administrator and PC chairpersons is to oversee the PCs. The PC can also be known as the reviewers. Some of the functionalities that the PC chairpersons would be able to execute include, enabling and disabling the discussion between reviewers, submission of reviews or submission of files, and to manage submitted papers from initial submission stage to the preparation of the conference proceeding.

The roles of PC members include choosing their preferences on submitted papers for reviews, to use a provided evaluation form to review submitted papers, and to discuss with other reviewers who have reviewed the same paper. The roles of authors are to submit papers, to review the reviews done on their paper and check the status of their paper, if it is accepted or rejected

Other functions that the PC chairperson may execute includes the ability to set the amount of papers a reviewer can receive, both globally and individually, assign paper reviewers and view a log of events of all users (except system administrators). The system will perform some roles automatically, such as assigning papers to the reviewers, set limits by default and to assign preferences on papers according to keywords and expertise of PCs when no preference is given.

The system cannot be completely automated or semi-automated, which can cause these limitations of the system:

- Disabling or enabling submission of papers must be done manually

- When the paper is submitted, the PC chairperson has to decide when to disable to the discussion box.

## Definitions, Acronyms and Abbreviations

Refer to Glossary document.

## References

* Rational Unified Process, SRS template (rup\_srs.doc)

## Overview

The remaining sections of this document provide a general description, including characteristics of the users of this project, the product's software, and the functional and data requirements of the product.

Section 2: Overall Description. Describes the general factors that affects the system.

Section 3: Specific Requirements. Contains all software requirements the system must have to meet the client’s need.

# Overall Description

## 2.1 Product perspective

Currently, all activities carried out under the management of the conference is done manually. The manual execution of some processes such as submission of a paper is inefficient and requires a large amount of time and man power to complete.

To overcome the inefficiency, the need to build a Conference Management System is raised which will effectively assist the organizers in managing the conference, simplifying processes such as submission of papers and assignment of papers to reviewers.

### System interface

The conference management system to be developed will be done on command line and can be deployed on the internet. Clients can simultaneously log into the system from any computer that supports internet services and interact with the system for their purposes, such as submission of papers, modifying account details and much more.

The cloud server allows client machines to connect to the Conference Management System and serve their needs via a command line program. It acts as a mechanism to accept and process client requests, retrieve data from text files and return the results to the requests.

### User interfaces

The user interfaces provided to client machines are based on command lines and can be accessed through any Windows machine. All users would be using the same user interface. Different roles of users would be able to view different interfaces after logging in into the system.

Accessing to text files containing data about users or papers must be performed indirectly though the command line program provided by the Conference Management System.

### Hardware interfaces

All components must be able to execute on a personal computer.

### Memory Constraints

The client machines must at least have 100mb of free space.

### Operations

The conference Management system must be easy for all users to use. In other words, no specific information or technical skills are required to use the system.

Files which stores data about the system should be simple enough for the network administrator to access and would not require any information or technical skills.

Backup and recovery operations must be specified in case of file corruption, power outage or any other incidents which may disrupt the system.

## Product Functions

The nine main functions of the Conference Management system are listed as below and categorized based on different roles of users that will be involved with it.

For Program Committee Chairs, the Conference Management System would provide the following functionalities to manage and monitor submission of papers, assignment of papers to reviewers and users:

* management and monitoring of the program committee;
* sophisticated and flexible management of the access of PC members to papers and conflicts of interests;
* paper assignment based on the preferences of PC members;
* list of the latest events;
* automatic preparation of conference proceedings

For Program Committee, the Conference Management system would provide functionalities such as to manage reviews and specify their preferences on submitted papers;

* submission of reviews;
* discussion of reviewing of papers;

For authors, the Conference Management system would provide functionalities:

* paper submission;
* the author response to the reviews given to their paper.

Functionalities listed here will be gone through in more detail in the next part. Each of the functionalities would be subdivided into smaller parts. Functionalities of the system are built based on the needs of a Conference Management system so that all activities and tasks carried out by the organizers are simplified and more convenient.

## User Characteristics

The users of the Conference Management system include administrators, Program Committee Chair, Program Committee and authors.

* Administrators have strong knowledge on how a conference management system is supposed to run and will maintain the system
* Program Committee Chair will enable/disable functionalities of the system which will be used by the Program Committee and authors. They also oversee the events in the system
* Program Committee will review papers that are assigned to them
* Authors submit papers to the system

## Constraints

The system should obey and satisfy the following constraints

* Backup and recovery: The backup and recovery of the system files must be easy to perform to prevent file corruptions and loss risks
* Access control: the system must provide appropriate access rights to different users of different roles
* Authentication: the system must check the username and password of users when logging in
* The system must be developed by November of 2014
* The system must be developed in C++ programming language and run on Windows machines.

## Assumptions and dependencies

All potential users of the system must have an email address.

# Specific Requirements

Each requirement specified (either functional or non-functional) of the Conference Management System is classified based on client’s classification of mandatory or optional.

* Mandatory: Requirements of highest importance level. Mandatory requirements are those that reflect core functionalities of the Conference Management System and must be implemented.
* Optional: Requirements of lowest importance level. Optional requirements reflect enhanced functionalities of the Conference Management System that should only be considered when all Mandatory requirements have already been completed.

## 3.1 Functional Requirements

**3.1.1 Users Management Subsystem**

This section captures functionalities that the Conference Management System provides for users to execute a variety of actions depending on their roles. Different roles include Administrator, Program Committee Chairs, Program Committee and normal users.

*3.1.1.1 Administrator Side*

This section includes functions for the system admin to manage the configuration of the system and assign different roles to the users.

|  |  |  |
| --- | --- | --- |
| **Requirement #:** F\_3.1.1.1\_01 | **Requirement Type:** Functional | **Use Case ID**: 27 |
| **Description:** The system should allow the administrator to assign Program Committee Chairs. | | |
| **Rationale:** System Admin wants to assign Program Committee Chairs. | | |
| **Source:** Client – Hoa Dam | | |
| **Fit Criterion:** System Admin successfully assigns Program Committee Chairs. | | |
| **Dependencies:** None. | | |
| **Classification:** Mandatory | | |
| **History:** Created by Ng Shien Wee on 29/8/2014 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement #:** F\_3.1.1.1\_02 | **Requirement Type:** Functional | **Use Case ID**: 28 |
| **Description:** The system should allow the administrator to create conferences. The system allows more than 1 conference to be created. | | |
| **Rationale:** System Admin wants to create a conference. | | |
| **Source:** Client – Hoa Dam | | |
| **Fit Criterion:** System Admin successfully creates a new conference. | | |
| **Dependencies:** None. | | |
| **Classification:** Mandatory | | |
| **History:** Created by Ng Shien Wee on 10/9/2014 | | |

*3.1.1.2 Program Committee Chair Side*

This section includes functions which allows the Program Committee (PC) Chairs to oversee and manage users of the system

|  |  |  |
| --- | --- | --- |
| **Requirement #:** F\_3.1.1.2\_01 | **Requirement Type:** Functional | **Use Case ID:** 12 |
| **Description:** The system should allow the PC chairs to assign Program Committees. By going through the list of users, assignment of Program Committees are based on the users interests and how closely tied the user’s interest is to the conference topic. | | |
| **Rationale:** PC chairs needs to assign Program Committees | | |
| **Source:** Client – Hoa Dam | | |
| **Fit Criterion:** PC chairs successfully assign a normal user the role of Program Committee | | |
| **Dependencies:** Users who are being assigned the role and the PC chair must have an existing account. Refer to F\_3.1.1.3\_01. | | |
| **Classification:** Mandatory | | |
| **History:** Created by Ng Shien Wee on 29/8/2014 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement #:** F\_3.1.1.2\_02 | **Requirement Type:** Functional | **Use Case ID:** 13 |
| **Description:** The system should allow the PC chairs to monitor the program committee. Types of information required when monitoring the program committee includes   * Number of papers assigned to each Program Committee * Papers assigned to each PC * Papers which has been reviewed by each PC * Papers which has not been reviewed by each PC * Reviews on paper wrote by each PC * Preferences on each paper | | |
| **Rationale:** PC chairs want to monitor the Program Committees. | | |
| **Source:** Client – Hoa Dam | | |
| **Fit Criterion:** PC chairs are able to monitor the Program Committee successfully. | | |
| **Dependencies:** PC chair and program committee both must have an existing account. Refer to F\_3.1.1.3\_01. | | |
| **Classification:** Mandatory | | |
| **History:** Created by Ng Shien Wee on 29/8/2014 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement #:** F\_3.1.1.2\_03 | **Requirement Type:** Functional | **Use Case ID:** 16 |
| **Description:** The system should allow the PC Chairs to check on the latest events. The type of events include   * Submission of a paper by an author * An upcoming deadline * Submission of review by a reviewer * An event that recently expired * Creation of new accounts | | |
| **Rationale:** PC chairs want to check out the latest events. | | |
| **Source:** Client – Hoa Dam | | |
| **Fit Criterion:** PC chairs are able to view the list of latest events successfully. | | |
| **Dependencies:** None | | |
| **Classification:** Mandatory | | |
| **History:** Created by Ng Shien Wee on 29/8/2014 | | |

*3.1.1.3 User side*

This section includes functions which allow users to interact with the system.

|  |  |  |
| --- | --- | --- |
| **Requirement #:** F\_3.1.1.3\_01 | **Requirement Type:** Functional | **Use Case ID:** None |
| **Description:** The system should allow creation of new accounts by new users. Types of information required when creating a new account includes   * First Name * Last Name * User Name * Email * Password * Expertise * University | | |
| **Rationale:** New users need to create a new account. | | |
| **Source:** None | | |
| **Fit Criterion:** New users successfully create a new account. | | |
| **Dependencies:** None | | |
| **Classification:** Mandatory | | |
| **History:** Created by Ng Shien Wee on 29/8/2014 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement #:** F\_3.1.1.3\_02 | **Requirement Type:** Functional | **Use Case ID:** None |
| **Description:** The system should allow users to login to their accounts. | | |
| **Rationale:** Users to login to the system. | | |
| **Source:** None | | |
| **Fit Criterion:** Users successfully login into their accounts. | | |
| **Dependencies:** Users who log in must have an existing account. Refer to F\_3.1.1.3\_01. | | |
| **Classification:** Mandatory | | |
| **History:** Created by Ng Shien Wee on 6/9/2014 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement #:** F\_3.1.1.3\_03 | **Requirement Type:** Functional | **Use Case ID:** 4 |
| **Description:** The system should allow users to change their own account passwords. | | |
| **Rationale:** Users to change their account password. | | |
| **Source:** None | | |
| **Fit Criterion:** Users successfully change their account password. | | |
| **Dependencies:** Users who modify their account passwords must have an existing account. Refer to F\_3.1.1.3\_01. | | |
| **Classification:** Mandatory | | |
| **History:** Created by Ng Shien Wee on 3/9/2014 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement #:** F\_3.1.1.3\_04 | **Requirement Type:** Functional | **Use Case ID:** 3 |
| **Description:** The system should allow users to modify their personal details. | | |
| **Rationale:** Users to modify their personal details. | | |
| **Source:** None | | |
| **Fit Criterion:** Users successfully modifies their personal details. | | |
| **Dependencies:** Users who modify their personal details must have an existing account. Refer to F\_3.1.1.3\_01. | | |
| **Classification:** Mandatory | | |
| **History:** Created by Ng Shien Wee on 3/9/2014 | | |

**3.1.2 Paper Management Subsystem**

This section captures the functionalities that the Conference Management System provides to the users to manage research papers, including submission of research papers, assignment of research papers to the reviewers.

*3.1.2.1 Paper submission*

This section includes functionalities for the users to submit papers to the system.

|  |  |  |
| --- | --- | --- |
| **Requirement #:** F\_3.1.2.1\_01 | **Requirement Type:** Functional | **Use Case ID:** 5 |
| **Description:** The system should allow existing users to submit papers to the system. User who submits a paper will automatically be assigned the role of author. During submission of papers, authors are to fill in forms about the details of the paper and also information about the author.   * Information of author includes  1. First Name 2. Last Name 3. Email 4. University  * Information of paper includes  1. Title 2. Abstract 3. Keywords   Authors who submit the paper will fill in details about other authors who have contributed to the paper. Other authors who have contributed to the paper must have an existing account. Authors are permitted to submit an unlimited amount of papers. | | |
| **Rationale:** Existing user needs to submit a paper to the system. | | |
| **Source:** Client – Hoa Dam | | |
| **Fit Criterion:** User successfully submits a paper to the system. | | |
| **Dependencies:** User who submits a paper must have an existing account. Other authors who have contributed to the paper must also have an existing account. Refer to F\_3.1.1.3\_01. | | |
| **Classification:** Mandatory | | |
| **History:**   * Created by Ng Shien Wee on 29/8/2014 * Edited by Ng Shien Wee on 3/9/2014 * Edited by Ng Shien Wee on 6/9/2014 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement #:** F\_3.1.2.1\_02 | **Requirement Type:** Functional | **Use Case ID:** 6 |
| **Description:** The system should allow authors to modify their submissions. Modification includes changing the details of the authors and resubmission of paper. This must be done before the submission deadline. | | |
| **Rationale:** Authors to modify their submissions. | | |
| **Source:** Client – Hoa Dam | | |
| **Fit Criterion:** Authors successfully modify their submissions. | | |
| **Dependencies:** User must have submitted a paper. Refer to F\_3.1.2.1\_01. | | |
| **Classification:** Mandatory | | |
| **History:**   * Created by Ng Shien Wee on 29/8/2014 * Edited by Ng Shien Wee on 6/9/2014 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement #:** F\_3.1.2.1\_03 | **Requirement Type:** Functional | **Use Case ID:** None |
| **Description:** The system should be able to assign users who have contributed to the papers automatically. This is related to requirement F\_3.1.2.1\_01. This process is only done when multiple author details has been added to a paper submission. Based on the email address of the authors (excluding the author who submitted the paper), the system should be able to link the paper to the respective user accounts. | | |
| **Rationale:** System to assign users to their papers. | | |
| **Source:** Client – Hoa Dam | | |
| **Fit Criterion:** System successfully assigns user to their papers. | | |
| **Dependencies:** When linking paper to user accounts, user account must exist. Refer to F\_3.1.1.3\_01. | | |
| **Classification:** Mandatory | | |
| **History:**   * Created by Ng Shien Wee on 29/8/2014 * Edited by Ng Shien Wee on 3/9/2014 | | |

*3.1.2.2 Paper assignment to reviewers*

This section includes functionalities for the Program Committees, which are also known as reviewers to be assigned a paper for review.

|  |  |  |
| --- | --- | --- |
| **Requirement #:** F\_3.1.2.2\_01 | **Requirement Type:** Functional | **Use Case ID:** 8 |
| **Description:** The system should allow reviewers to look through the list of paper submitted and specify their preferences on each paper. When looking through the list of papers, reviewers are only allowed to look at the following   * Keywords * Abstract * Title   Preferences that can be specified by the reviewers include “Yes”, “No”, “Maybe” and “Conflict of Interest”. | | |
| **Rationale:** Reviewers specify their preference on each paper. | | |
| **Source:** Client – Hoa Dam | | |
| **Fit Criterion:** Reviewers successfully specify their preference on each paper. | | |
| **Dependencies:** Reviewers must have an existing account. Refer to F\_3.1.1.3\_01. | | |
| **Classification:** Mandatory | | |
| **History:** Created by Ng Shien Wee on 29/8/2014 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement #:** F\_3.1.2.2\_02 | **Requirement Type:** Functional | **Use Case ID:** None |
| **Description:** The system should allow PC chairs to automate the process of specifying user preferences on papers. This process is only done if a user has not specified any preferences on a paper. This automation process involves looking through the keywords of the paper submitted with the expertise of the reviewer. A percentage value would be given on how closely related they are. Closely related being how many keywords match the expertise of the reviewer. The higher the percentage, the more closely related they are. | | |
| **Rationale:** System needs to automatically assign preferences to paper. | | |
| **Source:** Client – Hoa Dam | | |
| **Fit Criterion:** System successfully assigns preferences to paper. | | |
| **Dependencies:** None. | | |
| **Classification:** Mandatory | | |
| **History:** Created by Ng Shien Wee on 29/8/2014 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement #:** F\_3.1.2.2\_03 | **Requirement Type:** Functional | **Use Case ID:** 14 |
| **Description:** The system should allow PC chairs to automate the process of assigning papers to the reviewers. This process involves looking at the preferences (Preferences being “Yes”, “No”, “Maybe” or “Conflict of Interest”) of the reviewer and assigns the paper to him/her. There is a possibility where a reviewer can be assigned a paper where his/her preference is “No”. This happens under a certain condition being   * A paper has reached its maximum number of reviewers * A reviewer has not been assigned the maximum number of papers. | | |
| **Rationale:** System needs to assign papers to users | | |
| **Source:** Client – Hoa Dam | | |
| **Fit Criterion:** System successfully assign papers to users. | | |
| **Dependencies:** None | | |
| **Classification:** Mandatory | | |
| **History:** Created by Ng Shien Wee on 29/8/2014 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement #:** F\_3.1.2.2\_04 | **Requirement Type:** Functional | **Use Case ID:** 15 |
| **Description:** The system should allow PC Chairs to manually assign paper to reviewers. The automation process may have assigned the paper to reviewers who have a conflict of interest towards the paper. | | |
| **Rationale:** PC Chairs need to manually assign paper to reviewers. | | |
| **Source:** Client – Hoa Dam | | |
| **Fit Criterion:** PC Chairs successfully assign paper to reviewers. | | |
| **Dependencies:** None | | |
| **Classification:** Mandatory | | |
| **History:** Created by Ng Shien Wee on 29/8/2014 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement #:** F\_3.1.2.2\_05 | **Requirement Type:** Functional | **Use Case ID:** 21 |
| **Description:** The system should allow PC Chairs to set the number of reviewers a paper can receive. The number is dependent on the number of reviewers available and the amount of papers assigned. Default is set at 5. | | |
| **Rationale:** PC Chairs need to set the number of reviewers per paper. | | |
| **Source:** Client – Hoa Dam | | |
| **Fit Criterion:** PC Chairs successfully set the number of reviewers per paper. | | |
| **Dependencies:** None | | |
| **Classification:** Mandatory | | |
| **History:**   * Created by Ng Shien Wee on 29/8/2014 * Edited by Ng Shien Wee on 6/9/2014 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement #:** F\_3.1.2.2\_06 | **Requirement Type:** Functional | **Use Case ID:** 17 |
| **Description:** The system should allow PC Chairs to set the number of papers reviewers can receive. The number set by the PC chair is dependent on the number of papers submitted and the number of reviewers available. Default is set at 3. | | |
| **Rationale:** PC Chairs need to set a global limit for the number of papers all reviewers receive. Global limit being the number of papers all reviewers receive. | | |
| **Source:** Client – Hoa Dam | | |
| **Fit Criterion:** PC Chairs successfully set a global limit for it for the number of papers all reviewers receive. | | |
| **Dependencies:** None | | |
| **Classification:** Mandatory | | |
| **History:**   * Created by Ng Shien Wee on 29/8/2014 * Edited by Ng Shien Wee on 6/9/2014 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement #:** F\_3.1.2.2\_07 | **Requirement Type:** Functional | **Use Case ID:** 18 |
| **Description:** The system should also allow manual assignment of the number of papers a reviewer can receive. Some reviewers may be busier than the others and assigned lesser amount of papers for reviewing or vice versa. | | |
| **Rationale:** PC Chairs need to manually assign the number of papers a reviewer can receive. | | |
| **Source:** Client – Hoa Dam | | |
| **Fit Criterion:** PC Chairs successfully assign the number of papers a reviewer can receive. | | |
| **Dependencies:** None | | |
| **Classification:** Mandatory | | |
| **History:**   * Created by Ng Shien Wee on 7/9/2014 | | |

*3.1.2.3 Conference Proceeding*

This section includes functionalities for the preparation of conference proceedings.

|  |  |  |
| --- | --- | --- |
| **Requirement #:** F\_3.1.2.3\_01 | **Requirement Type:** Functional | **Use Case ID:** 20 |
| **Description:** The system should allow the PC Chairs to automate the process to generate the conference proceeding. This is done by looking through the list of papers, then generating a list of accepted papers. List of accepted papers are grouped according to the topics. | | |
| **Rationale:** System automatically generates conference proceeding. | | |
| **Source:** Client – Hoa Dam | | |
| **Fit Criterion:** System successfully generates the conference proceeding. | | |
| **Dependencies:** None | | |
| **Classification:** Mandatory | | |
| **History:** Created by Ng Shien Wee on 29/8/2014 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement #:** F\_3.1.2.3\_02 | **Requirement Type:** Functional | **Use Case ID:** 26 |
| **Description:** The system should allow Program Committee Chairs to manually group the list of papers based on the topics for the conference proceeding. The automation process may have grouped the paper incorrectly. | | |
| **Rationale:** PC chairs manually group papers based on their topics. | | |
| **Source:** Client – Hoa Dam | | |
| **Fit Criterion:** PC Chairs successfully group papers based on the topics. | | |
| **Dependencies:** None | | |
| **Classification:** Mandatory | | |
| **History:** Created by Ng Shien Wee on 29/8/2014 | | |

**3.1.3 Paper Review Management Subsystem**

This section captures the functionalities that the Conference Management System provides for users to execute different actions related to paper reviews. Different roles include Program Committee Chairs, Program Committee and the authors.

*3.1.3.1 Program Committee Chairs Side*

This section includes functionalities for the Program Committee Chairs to manage paper reviews.

|  |  |  |
| --- | --- | --- |
| **Requirement #:** F\_3.1.3.1\_01 | **Requirement Type:** Functional | **Use Case ID:** 25 |
| **Description:** The system should allow Program Committee Chairs to enable or disable paper review discussions among the reviewers. | | |
| **Rationale:** Program Committee Chair to enable or disable review discussions. | | |
| **Source:** Client – Hoa Dam | | |
| **Fit Criterion:** Program Committee Chairs successfully enables or disable review discussions. | | |
| **Dependencies:** None | | |
| **Classification:** Mandatory | | |
| **History:** Created by Ng Shien Wee on 29/8/2014 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement #:** F\_3.1.3.1\_02 | **Requirement Type:** Functional | **Use Case ID:** 19 |
| **Description:** The system should allow Program Committee Chairs to accept or reject a paper based on the reviews and discussion submitted by the reviewers. PC chairs will have a page listing all the papers submitted, choose a paper and get more details about the paper. Details about the paper include   * Information about the contributed authors * Paper information * Reviewers * Completed Reviews * Discussions | | |
| **Rationale:** Program Committee Chair to enable or disable review discussions. | | |
| **Source:** Client – Hoa Dam | | |
| **Fit Criterion:** Program Committee Chairs successfully enables or disable review discussions. | | |
| **Dependencies:** None | | |
| **Classification:** Mandatory | | |
| **History:** Created by Ng Shien Wee on 29/8/2014 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement #:** F\_3.1.3.1\_03 | **Requirement Type:** Functional | **Use Case ID:** 24 |
| **Description:** The system should allow Program Committee Chairs to enable or disable the functionality where the authors are able to leave a respond about the reviews on his/her paper. | | |
| **Rationale:** Program Committee Chair to enable or disable author response on the review. | | |
| **Source:** Client – Hoa Dam | | |
| **Fit Criterion:** Program Committee Chairs successfully enables or disable author response on the review. | | |
| **Dependencies:** None | | |
| **Classification:** Mandatory | | |
| **History:** Created by Ng Shien Wee on 29/8/2014 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement #:** F\_3.1.3.1\_04 | **Requirement Type:** Functional | **Use Case ID:** 23 |
| **Description:** The system should allow Program Committee Chairs to enable or disable paper review submissions by the Program Committee. | | |
| **Rationale:** Program Committee Chair to enable or disable review submissions. | | |
| **Source:** Client – Hoa Dam | | |
| **Fit Criterion:** Program Committee Chairs successfully enables or disable review submissions. | | |
| **Dependencies:** None | | |
| **Classification:** Mandatory | | |
| **History:** Created by Ng Shien Wee on 6/9/2014 | | |

*3.1.3.2 Reviewers Side*

This section includes functionalities for the Program Committee to submit paper reviews and discuss reviews done by other reviewers.

|  |  |  |
| --- | --- | --- |
| **Requirement #:** F\_3.1.3.2\_01 | **Requirement Type:** Functional | **Use Case ID:** 9 |
| **Description:** The system should allow the reviewers to choose from a list of assigned papers and review the paper based on the given evaluation form. The reviewer submits the review when he/she has completed the review. | | |
| **Rationale:** Reviewers to submit paper reviews. | | |
| **Source:** Client – Hoa Dam | | |
| **Fit Criterion:** Reviewers successfully submits the paper review. | | |
| **Dependencies:** None | | |
| **Classification:** Mandatory | | |
| **History:** Created by Ng Shien Wee on 29/8/2014 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement #:** F\_3.1.3.2\_02 | **Requirement Type:** Functional | **Use Case ID:** 10 |
| **Description:** The system should allow the reviewers to modify the reviews he/she has submitted. | | |
| **Rationale:** Reviewers to modify paper reviews | | |
| **Source:** Client – Hoa Dam | | |
| **Fit Criterion:** Reviewers successfully modifies the paper review. | | |
| **Dependencies:** The reviewer must have submitted a paper review to modify it. Refer to F\_3.1.3.2\_01 | | |
| **Classification:** Mandatory | | |
| **History:** Created by Ng Shien Wee on 29/8/2014 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement #:** F\_3.1.3.2\_02 | **Requirement Type:** Functional | **Use Case ID:** 11 |
| **Description:** The system should allow reviewers to discuss about the reviews submitted by different reviewers, who have been assigned the same paper. Authors are not allowed to see the discussion among the reviewers. | | |
| **Rationale:** Reviewers to discuss with other reviewers about the paper reviews | | |
| **Source:** Client – Hoa Dam | | |
| **Fit Criterion:** Reviewers successfully discuss with other reviewers about their paper reviews. | | |
| **Dependencies:** None | | |
| **Classification:** Mandatory | | |
| **History:** Created by Ng Shien Wee on 29/8/2014 | | |

*3.1.3.3 Authors Side*

This section includes functionalities for authors who have submitted papers to respond to reviews provided to them.

|  |  |  |
| --- | --- | --- |
| **Requirement #:** F\_3.1.3.3\_01 | **Requirement Type:** Functional | **Use Case ID:** 7 |
| **Description:** The system should allow authors to respond to reviews on their paper. This is also known the rebuttal phase. Authors are allowed to read the reviews and leave a comment. | | |
| **Rationale:** Authors are able to leave a comment regarding the reviewers review. | | |
| **Source:** Client – Hoa Dam | | |
| **Fit Criterion:** Authors successfully leaves a comment. | | |
| **Dependencies:** User must have submitted a paper. Refer to F\_3.1.2.1\_01 | | |
| **Classification:** Mandatory | | |
| **History:** Created by Ng Shien Wee on 29/8/2014 | | |

3.2 Non-Functional Requirement

|  |  |  |
| --- | --- | --- |
| **Requirement #:** NFREQ01 | **Requirement Type:** Performance | **Use Case #:** None |
| **Description:** The system should be able to work 24/7 except during maintenance. | | |
| **Rationale:** All users want to access the system 24/7. | | |
| **Source:** All users who interact with the system | | |
| **Fit Criterion:** Authorized users can use the system at any time (24/7). | | |
| **Dependencies:** None | | |
| **Classification:** Mandatory | | |
| **History:** Created by Ng Shien Wee and Girit on 3/9/2014 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement #:** NFREQ02 | **Requirement Type:** Performance | **Use Case #:** None |
| **Description:** The system should be responsive. The system must respond to actions by any users within 20 seconds. | | |
| **Rationale:** All users want to receive a quick respond from the system. | | |
| **Source:** All users who interact with the system | | |
| **Fit Criterion:** Response by the system towards the user’s action is given within 20 seconds. | | |
| **Dependencies:** None | | |
| **Classification:** Mandatory | | |
| **History:** Created by Ng Shien Wee and Girit on 3/9/2014 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement #:** NFREQ03 | **Requirement Type:** Performance | **Use Case #:** None |
| **Description:** The system should generate the conference proceeding within no more than 1 minute. | | |
| **Rationale:** The PC chairs wants to view the conference proceeding as quickly as possible. | | |
| **Source:** PC chairs | | |
| **Fit Criterion:** The conference proceeding is generated within 1 minute. | | |
| **Dependencies:** None | | |
| **Classification:** Mandatory | | |
| **History:** Created by Ng Shien Wee and Girit on 3/9/2014 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement #:** NFREQ04 | **Requirement Type:** Performance | **Use Case #:** None |
| **Description:** The system should display details about an author and paper when prompted by the PC chairs within 20 seconds. | | |
| **Rationale:** The PC chairs want to obtain more information about the author or the paper as quickly as possible. | | |
| **Source:** PC chairs | | |
| **Fit Criterion:** The details about authors and the paper are displayed for the PC chair within 20 seconds. | | |
| **Dependencies:** None | | |
| **Classification:** Mandatory | | |
| **History:** Created by Ng Shien Wee and Girit on 3/9/2014 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement #:** NFREQ05 | **Requirement Type:** Usability | **Use Case #:** None |
| **Description:** The system should provide a simple “Help” page for new users or existing users at the main page of the system. The help page provides simple tips to learn more about the system and provide contact details so that the user can ask for more advanced help. | | |
| **Rationale:** To ease users who are new in using the system. | | |
| **Source:** All users who interact with the system. | | |
| **Fit Criterion:** The help is accessible to all users of the system. | | |
| **Dependencies:** None | | |
| **Classification:** Optional | | |
| **History:** Created by Ng Shien Wee and Girit on 3/9/2014 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement #:** NFREQ06 | **Requirement Type:** Usability | **Use Case #:** None |
| **Description:** The system should be easy to use for existing users or new users. It should not require any specific knowledge or technical skills. No additional training should be provided for the users so that within 2 days of exploring the system, the user should have a good overview of the system. | | |
| **Rationale:** User wants to use the system without going through any additional training. | | |
| **Source:** All users who interact with the system. | | |
| **Fit Criterion:** The user is able to use the Conference Management System within a 2 days period. | | |
| **Dependencies:** None | | |
| **Classification:** Mandatory | | |
| **History:** Created by Ng Shien Wee and Girit on 3/9/2014 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement #:** NFREQ07 | **Requirement Type:** Usability | **Use Case #:** None |
| **Description:** The system should run on command line and in English. | | |
| **Rationale:** User wants to use a familiar language when using the system. | | |
| **Source:** All users who interact with the system. | | |
| **Fit Criterion:** The system runs on command line and in English. | | |
| **Dependencies:** None | | |
| **Classification:** Mandatory | | |
| **History:** Created by Ng Shien Wee and Girit on 3/9/2014 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement #:** NFREQ08 | **Requirement Type:** Reliability | **Use Case #:** None |
| **Description:** The system should recover from failure no more than 2 days. | | |
| **Rationale:** System to recover from a failure as fast as possible. | | |
| **Source:** All users who interact with the system. | | |
| **Fit Criterion:** The system not taking more than 2 days to recover. | | |
| **Dependencies:** None | | |
| **Classification:** Mandatory | | |
| **History:** Created by Ng Shien Wee and Girit on 3/9/2014 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement #:** NFREQ09 | **Requirement Type:** Reliability | **Use Case #:** None |
| **Description:** The system should support more than 30 simultaneous users at a time. | | |
| **Rationale:** To support multiple users to use the system. | | |
| **Source:** All users who interact with the system. | | |
| **Fit Criterion:** The system should support more than 30 users at a time. | | |
| **Dependencies:** None | | |
| **Classification:** Mandatory | | |
| **History:** Created by Ng Shien Wee and Girit on 3/9/2014 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement #:** NFREQ10 | **Requirement Type:** Security | **Use Case #:** None |
| **Description:** The system should automatically log out after 10 minutes of idle time. | | |
| **Rationale:** To control the amount of users who are currently logged on to the system. | | |
| **Source:** All users who interact with the system. | | |
| **Fit Criterion:** The system logs out a user who has been idle for more than 10 minutes. | | |
| **Dependencies:** None | | |
| **Classification:** Optional | | |
| **History:** Created by Ng Shien Wee and Girit on 3/9/2014 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement #:** NFREQ11 | **Requirement Type:** Performance | **Use Case #:** None |
| **Description:** The system should be able to store up to 200 user profiles and 200 different papers submitted by users. | | |
| **Rationale:** System to store as many papers and user profiles as possible. | | |
| **Source:** All users who interact with the system. | | |
| **Fit Criterion:** 200 user profiles and 200 different papers are stored in the system | | |
| **Dependencies:** None | | |
| **Classification:** Optional | | |
| **History:** Created by Ng Shien Wee and Girit on 3/9/2014 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement #:** NFREQ12 | **Requirement Type:** Usability | **Use Case #:** None |
| **Description:** The system should have user documentation and manuals to instruct users on how to use the Conference Management System. | | |
| **Rationale:** To shorten the amount of time needed to get used to the system | | |
| **Source:** All users who interact with the system. | | |
| **Fit Criterion:** The documentation helped the users understand better of the system. | | |
| **Dependencies:** None | | |
| **Classification:** Optional | | |
| **History:** Created by Ng Shien Wee and Girit on 3/9/2014 | | |

**Use Case**

**Use Case Description**

|  |  |
| --- | --- |
| **Name:** Create Account | **ID:** 1 |
| **Stakeholders and Goals:** New user – to be registered to the system | |
| **Description:** A new user wants to register to use the system to take part in the conference. | |
| **Actors:** New user | |
| **Trigger:** User gets to the main page of the system and chooses the option to register. | |
| **Normal Flow:**   1. New user reaches the front page of the system and chooses to register. 2. The new user would fill in the registration form with correct and up-to-date information about them. 3. The system then checks the user name entered by the new user against the list of user names which currently exist in the system. 4. If the user name entered is unique, the system displays that the account has been successfully created and can now login into the system. 5. Steps 1 to 4 are repeated for each new account. 6. End | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:**  4a. If the username entered is not unique, the system prompts the user for a new user name. | |

|  |  |
| --- | --- |
| **Name:** Login | **ID:** 2 |
| **Stakeholders and Goals:** Existing user – wants to login into the system | |
| **Description:** An existing user wants to login into the system. | |
| **Actors:** Author | |
| **Trigger:** User gets to the front page and chooses the option to login. | |
| **Normal Flow:**   1. The existing user enters his/her username and password in the login page. 2. The system checks if the username exists then check the password to see if it matches. 3. The system then displays the main page of the system, where it lists down the things that the user can do. 4. End | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:**  2a. If the username and password does not exist or match, the system prompts the user for his/her username or password again. | |

|  |  |
| --- | --- |
| **Name:** Modify personal details | **ID:** 3 |
| **Stakeholders and Goals:** Existing user – wants to modify their personal details | |
| **Description:** An existing user wants to modify their personal details. | |
| **Actors:** Author | |
| **Trigger:** After logging in, user attempts to modify their personal details. | |
| **Normal Flow:**   1. The user chooses “Modify personal details” at the main page and is brought to the page where it displays the details of the user. 2. The user then chooses which of his/her personal details to modify. 3. The system prompts for input from the user about the attribute to modify. 4. The user submits after entering the new details. 5. The system displays to the user that his/her detail has been updated. 6. Steps 1 to 5 are repeated if the user wishes to modify more about his/her personal details. 7. End | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:** None | |

|  |  |
| --- | --- |
| **Name:** Modify password | **ID:** 4 |
| **Stakeholders and Goals:** Existing user – wants to modify their account password | |
| **Description:** An existing user wants to modify their account password | |
| **Actors:** Author | |
| **Trigger:** After logging in, user attempts to modify their account password. | |
| **Normal Flow:**   1. The user chooses “Modify account password” at the main page and is brought to the page where it will allow the user to change their password. 2. The system prompts the user for the new password 3. The user submits the new password. 4. The system prompts for reconfirmation to change their account password. 5. User accepts and the system displays that the password has been successfully modified. 6. End | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:** None | |

|  |  |
| --- | --- |
| **Name:** Submit papers | **ID:** 5 |
| **Stakeholders and Goals:** Existing user – wants to submit a new paper | |
| **Description:** An existing user wants to submit a paper into the system. | |
| **Actors:** Author | |
| **Trigger:** After logging in, user gets to the new paper submission page and attempts to submit a paper. | |
| **Normal Flow:**   1. The user chooses “Submit a new paper” at the main page and is brought to the paper submission page. 2. The user then enters information about the author who contributed to the paper and also details about the paper. 3. If there are multiple authors who contributed to the paper, the user chooses the option to add more authors. New authors are added one at a time. 4. The system then checks the email of the new author and attempts to link the email address to an existing user in the system. 5. If the system finds a match to the email address, and successfully linked it to an existing user, the system then displays to the user that he/she has successfully added a new author. 6. The user then submits the paper when he/she has completed filling up the form. 7. The system displays to the user he/she has successfully submitted the paper and brings the user back to the main page. 8. Steps 3 to 5 are repeated if there are more than 2 authors who contributed to the paper. 9. Steps 1 to 7 are repeated if there are multiple papers that the user wishes to submit. 10. End | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:**  4a. If the email address entered about the new author does not exist; the system does not allow the user to add the author into the list of contributed authors. | |

|  |  |
| --- | --- |
| **Name:** Modify paper submission | **ID:** 6 |
| **Stakeholders and Goals:** Existing user – wants to modify a paper submission | |
| **Description:** An author wishes to modify their paper submissions. | |
| **Actors:** Author | |
| **Trigger:** After logging in, author gets to the modify paper submission page and attempts to modify their paper submission. | |
| **Normal Flow:**   1. The user chooses “Modify paper submission” at the main page and is brought to the modify paper submission page. 2. The system displays to the user the papers that he has contributed to and provides the user with choices on what to modify. Choices include modify details about the authors, modify details about the paper or resubmit a paper. 3. The user makes a selection. 4. The system prompts for the new details or prompts for the new file submission. 5. The user submits the new changes. 6. The system displays to the user that the changes has been made to the paper submission. 7. Steps 1 to 6 is repeated if there are more changes to the paper submission. 8. End. | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:**  6a. If the user is modifying the author details and the new email address is entered about the author and does not exist; the system does not allow the user to add the author into the list of contributed authors. | |

|  |  |
| --- | --- |
| **Name:** Respond to reviews | **ID:** 7 |
| **Stakeholders and Goals:** Author – wants to respond to reviews | |
| **Description:** An author wishes to respond to the reviews given to his/her paper by the reviewer | |
| **Actors:** Author | |
| **Trigger:** After logging in, author gets to the submitted paper page and is able to see the reviews provided. The author then wishes to leave a comment about the reviews given | |
| **Normal Flow:**   1. The author chooses the “Papers submitted” page and is brought to the page where it lists all the paper submitted by the user. 2. The user then chooses a paper. 3. The system then displays the details about the paper and also the reviews given to the paper. 4. The author reads through the review and chooses the option to respond to the review. 5. The system prompts for the author’s comment. 6. The author submits the comment upon completion. Author is not allowed to submit the comment more than once. 7. The system displays that the user has successfully submitted the comment. 8. If the author has contributed to more than 1 paper and wishes to respond to more reviews, step 1 to 7 is repeated. 9. End | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:** None | |

|  |  |
| --- | --- |
| **Name:** Specify preference | **ID:** 8 |
| **Stakeholders and Goals:** Program Committee – wants to specify his/her preference on submitted papers. | |
| **Description:** A program committee wants to specify his/her preference on submitted papers so that he can be assigned papers of his interest | |
| **Actors:** Program Committee | |
| **Trigger:** After logging in, program committee gets to the page where he/she is able to specify his/her preference | |
| **Normal Flow:**   1. The Program Committee (PC) chooses the “Specify preference” page and is brought to the page. 2. The system lists all the papers currently submitted to the system. 3. The PC looks selects a paper. 4. The system displays the title of the paper, the abstract and the keywords to the PC. The system also displays the preferences available to the PC, which is “Yes”, “No”, “Maybe” or “Conflict of interest” and prompts the user for his/her preference. 5. The PC submits his preference on the paper. 6. The system then asks the PC if he/she wants to look through more papers to specify his/her preference. 7. Steps 2 to 6 are repeated is the PC wants to continuously specify preferences on the paper. 8. Steps 1 to 6 are repeated the next time the user comes back to the system to specify preferences on other papers. 9. End | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:**  6a. If there are no more papers, the system displays to the PC that there is no more papers left to specify his/her preference on and is brought back to the main page. | |

|  |  |
| --- | --- |
| **Name:** Review paper | **ID:** 9 |
| **Stakeholders and Goals:** Program Committee – wants to review papers | |
| **Description:** A program committee wants to review a paper that has been assigned to him/her. | |
| **Actors:** Program Committee | |
| **Trigger:** After logging in, program committee gets to the page where he/she can start reviewing papers. | |
| **Normal Flow:**   1. The Program Committee (PC) chooses the “Review papers” page. 2. The system lists the paper that the PC has been assigned and prompts for the PC’s choice on which paper to review. 3. The PC chooses a paper. 4. The system then displays the review criteria. 5. The PC specifies his/her review and submits. 6. The system displays that the criteria is completed 7. Steps 4 to 6 are repeated until all review criteria (based on the evaluation form given) is completed. 8. The system displays that the review for the paper is completed and is brought back to the main page. 9. Steps 1 to 8 are repeated if the PC wants to review more papers. 10. End | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:**  3a. If the PC keys in the wrong choice, the system will prompt again for user input. | |

|  |  |
| --- | --- |
| **Name:** Modify review | **ID:** 10 |
| **Stakeholders and Goals:** Program Committee – wants to modify a submitted review | |
| **Description:** A program committee wants to modify a paper review that has been submitted | |
| **Actors:** Program Committee | |
| **Trigger:** After logging in, program committee gets to the page where he/she can modify submitted reviews. | |
| **Normal Flow:**   1. The Program Committee (PC) chooses the “Modify reviews” page. 2. The system lists the paper that the PC has been assigned and prompts for the PC’s choice on which paper to modify the review. 3. The PC chooses a paper. 4. The system then displays the review that has already been submitted. The system then lets the PC to choose which criteria that he/she wishes to modify. 5. The PC enters the choice. 6. The system prompts for the new review. 7. The PC submits upon completion. 8. The system displays that the PC has successfully updated the review 9. Steps 1 to 8 are repeated if the PC wishes to modify more reviews. 10. End | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:**  5a. If the PC keys in the wrong choice, the system will prompt again for user input. | |

|  |  |
| --- | --- |
| **Name:** Discuss review | **ID:** 11 |
| **Stakeholders and Goals:** Program Committee – wants to discuss paper reviews | |
| **Description:** A program committee wants to discuss with other program committee about the reviews given on the assigned paper | |
| **Actors:** Program Committee | |
| **Trigger:** After logging in, program committee gets to the page where he/she can start discussing the reviews on the paper. | |
| **Normal Flow:**   1. The Program Committee (PC) chooses the “Discuss reviews” page. 2. The system lists the paper that the PC has been assigned and prompts for the PC’s choice on which paper to go to for discussion. 3. The PC chooses a paper. 4. The system then displays the reviews submitted by different PCs and prompts for comment. 5. The PC submits his/her comment. 6. The system displays that the comment has been successfully posted. 7. Steps 1 to 6 is repeated if the PC wishes to go into other paper discussions. 8. End | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:**  3a. If the PC keys in the choice wrong, the system will prompt again for user input. | |

|  |  |
| --- | --- |
| **Name:** Assign PC (Program Committee) | **ID:** 12 |
| **Stakeholders and Goals:** Program Committee Chair – wants to assign new PCs | |
| **Description:** A program committee chair wants to look through a list of users and assign new PCs | |
| **Actors:** Program Committee Chairs | |
| **Trigger:** After logging in, program committee chairs gets to the page where he/she can start assigning the role of “Program Committee” to normal users. | |
| **Normal Flow:**   1. The Program Committee Chairs chooses the “Assign Program Committee” page. 2. The system lists all the users in the system and their expertise. The system also prompts the PC chair for his/her choice on which user to be assigned the role “Program Committee”. 3. The PC chair submits his/her choice. 4. The system reconfirms with the PC chair to assign the role to the user. 5. The PC chair submits confirmation. 6. The system then displays that the user’s role has been successfully changed. 7. Steps 1 to 6 are repeated if the PC chair wishes to assign more users with the role of “Program Committee”. 8. End | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:**  3a. If the PC keys in the choice wrong, the system will prompt again for user input. | |

|  |  |
| --- | --- |
| **Name:** Monitor PC (Program Committee) | **ID:** 13 |
| **Stakeholders and Goals:** Program Committee Chair – wants to monitor PCs | |
| **Description:** A program committee chair wants to monitor the activities of the PCs. | |
| **Actors:** Program Committee Chairs | |
| **Trigger:** After logging in, program committee chairs gets to the page where he/she can monitor the PCs. | |
| **Normal Flow:**   1. The Program Committee Chairs chooses the “Monitor Program Committee” page. 2. The system lists all the PCs in the system and displays details about each PCs (Types of details can be seen in SRS). 3. The PC chair navigate through the list of PCs 4. The system displays page by page details about each PCs. 5. End | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:** None | |

|  |  |
| --- | --- |
| **Name:** Run paper assignment process | **ID:** 14 |
| **Stakeholders and Goals:** Program Committee Chair – run paper assignment process | |
| **Description:** A program committee chair wants to run the process that automatically assigns paper to reviewers. | |
| **Actors:** Program Committee Chairs | |
| **Trigger:** After logging in, program committee chairs gets to the page where he/she can enable or disable functionalities in the system. | |
| **Normal Flow:**   1. The Program Committee Chairs chooses the “Functionalities management” page. 2. The system lists all the functionalities that can be controlled by the PC chairs and prompts the PC chair on which he/she wishes to enable/disable. 3. The PC chair submits his/her choice, “Automatically assign papers to reviewers”. 4. The system prompts for reconfirmation. 5. The PC chair submits confirmation 6. The system displays that the paper assignment to reviewers is successful. 7. End | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:** None | |

|  |  |
| --- | --- |
| **Name:** Manual assignment of papers | **ID:** 15 |
| **Stakeholders and Goals:** Program Committee Chair – manually assign paper to reviewers | |
| **Description:** A program committee chair wants to manually assign papers to reviewers. | |
| **Actors:** Program Committee Chairs | |
| **Trigger:** After logging in, program committee chairs gets to the page where he/she can manually assign papers to reviewers. | |
| **Normal Flow:**   1. The Program Committee Chairs chooses the “Manually assign paper” page. 2. The system lists all the papers submitted to the system and prompts for choice. 3. The PC chairs chooses the paper. 4. The system then lists all the PC in the system and prompts for choice. 5. The PC chairs chooses the PC to be assigned the paper. 6. The system then displays the paper selected and the PC that is going to be assigned the paper and prompts for reconfirmation. 7. The PC chair confirms his/her selection 8. The system displays that the paper has been successfully assigned to the selected PC. 9. The steps 1 to 8 are repeated if there are more manual assignment of papers to PC. 10. End | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:** None | |

|  |  |
| --- | --- |
| **Name:** Check latest events | **ID:** 16 |
| **Stakeholders and Goals:** Program Committee Chair – check latest events | |
| **Description:** A program committee chair wants to check the latest happenings in the system. | |
| **Actors:** Program Committee Chairs | |
| **Trigger:** After logging in, program committee chairs gets to the page where he/she can see the latest events in the system. | |
| **Normal Flow:**   1. The Program Committee Chairs chooses the “Check latest events” page. 2. The system lists 20 of the latest events in the system (More details in the SRS). 3. To refresh the list of events, steps 1 and 2 is repeated 4. End | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:** None | |

\*\*\*

|  |  |
| --- | --- |
| **Name:** Set number of papers per reviewers | **ID:** 17 |
| **Stakeholders and Goals:** Program Committee Chair – set number of papers per reviewers | |
| **Description:** A program committee chair wants to set the number of paper each reviewer can receive. | |
| **Actors:** Program Committee Chairs | |
| **Trigger:** After logging in, program committee chairs gets to the page where he/she can set the number of papers per reviewers. | |
| **Normal Flow:**   1. The Program Committee Chairs chooses the “Functionalities management” page. 2. The system lists all the functionalities that can be controlled by the PC chairs and prompts the PC chair on which he/she wishes to enable/disable. 3. The PC chair chooses “Set number of papers each reviewers receive”. 4. The system prompts for a number from the PC chair. 5. The PC chair sets the number of reviewers per paper. 6. The system reconfirms with the PC chair 7. The PC confirms the number 8. The system displays that he/she has successfully changed the number of papers per reviewer. 9. End | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:** None | |

|  |  |
| --- | --- |
| **Name:** Manually set the number of papers a reviewer receive | **ID:** 18 |
| **Stakeholders and Goals:** Program Committee Chair – manually set number of papers a reviewers receive. | |
| **Description:** A program committee chair wants to manually set the number of paper a reviewer can receive. | |
| **Actors:** Program Committee Chairs | |
| **Trigger:** After logging in, program committee chairs gets to the page where he/she can set the number of papers per reviewers. | |
| **Normal Flow:**   1. The Program Committee Chairs chooses the “Manually set number of papers a reviewers receive.” page. 2. The system lists all the PC in the system and prompts for PC chair’s choice. 3. The PC chair submits his/her choice. 4. The system then prompts for the number of paper the reviewer should receive. 5. The PC chair submits the number to be assigned. 6. The system prompts for reconfirmation from the PC chair. 7. The PC confirms his/her decision 8. The system displays that the PC has been successfully assigned the number of papers 9. Steps 1 to 8 is repeated if the PC chair needs to assign more users with number of papers he/she can receive manually. 10. End | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:** None | |

|  |  |
| --- | --- |
| **Name:** Accept/Reject paper | **ID:** 19 |
| **Stakeholders and Goals:** Program Committee Chair – accept or reject submitted papers | |
| **Description:** A program committee chair wants to accept or reject a paper. | |
| **Actors:** Program Committee Chairs | |
| **Trigger:** After logging in, program committee chairs gets to the page where he/she can start accepting or reject papers. | |
| **Normal Flow:**   1. The Program Committee Chairs chooses the “Accept/Reject papers” page. 2. The system lists all the submitted papers in the system. The system prompts the PC chair to select a paper. 3. The PC chair submits his/her choice. 4. The system lists the details about the paper which includes the author details, paper details and the reviews. The system then prompts the PC chair for his decision whether to accept or reject the paper. 5. The PC chair submits his/her decision 6. The system reconfirms the PC chair’s decision. 7. The PC chair confirms his/her decision. 8. The system displays that the PC chair’s decision about the paper has been successfully stored in the system. 9. The system prompts the PC chair if he/she wants to review the next paper. 10. Steps 2 to 9 are repeated if the PC chair wishes to continue reviewing. 11. Steps 1 to 10 are repeated if the PC chair wishes to repeat the reviewing process from the beginning. 12. End | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:**  9a. If all papers has been reviewed, the system displays that all papers has been reviewed and goes back to the main page. | |

|  |  |
| --- | --- |
| **Name:** Generate conference proceeding | **ID:** 20 |
| **Stakeholders and Goals:** Program Committee Chair – generate conference proceeding | |
| **Description:** A program committee chair wants to automate the process of generating the conference proceeding. | |
| **Actors:** Program Committee Chairs | |
| **Trigger:** After logging in, program committee chairs gets to the page where he/she can enable or disable functionalities in the system. | |
| **Normal Flow:**   1. The Program Committee Chairs chooses the “Functionalities management” page. 2. The system lists all the functionalities that can be controlled by the PC chairs and prompts the PC chair on which he/she wishes to enable/disable. 3. The PC chair chooses “Generate conference proceeding” and submits his/her choice. 4. The system reconfirms the PC chair’s decision to generate the conference proceeding 5. The PC chair confirms his decision 6. The system displays that the conference proceeding has been successfully generated. 7. End | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:**  6a. If there are no papers submitted, the system displays that there are no papers submitted and goes back to the main page | |

|  |  |
| --- | --- |
| **Name:** Set number of reviewers per paper | **ID:** 21 |
| **Stakeholders and Goals:** Program Committee Chair – set number of reviewers per paper | |
| **Description:** A program committee chair wants to set the number of number of reviewers each paper can receive. | |
| **Actors:** Program Committee Chairs | |
| **Trigger:** After logging in, program committee chairs gets to the page where he/she can enable or disable functionalities in the system. | |
| **Normal Flow:**   1. The Program Committee Chairs chooses the “Functionalities management” page. 2. The system lists all the functionalities that can be controlled by the PC chairs and prompts the PC chair on which he/she wishes to enable/disable. 3. The PC chair chooses “Generate conference proceeding” and submits his/her choice. 4. The system reconfirms the PC chair’s decision to generate the conference proceeding 5. The PC chair confirms his decision 6. The system displays that the conference proceeding has been successfully generated. 7. End | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:**  6a. If there are no papers submitted, the system displays that there are no papers submitted and goes back to the main page | |

|  |  |
| --- | --- |
| **Name:** Run specifying preference process | **ID:** 22 |
| **Stakeholders and Goals:** Program Committee Chair – automatically specify PC preference | |
| **Description:** A program committee chair wants to run the automation process of specifying preference for the PC who has not given his/her preference on a paper. | |
| **Actors:** Program Committee Chairs | |
| **Trigger:** After logging in, program committee chairs gets to the page where he/she can enable or disable functionalities in the system. | |
| **Normal Flow:**   1. The Program Committee Chairs chooses the “Functionalities management” page. 2. The system lists all the functionalities that can be controlled by the PC chairs and prompts the PC chair on which he/she wishes to enable/disable. 3. The PC chair chooses “Automatically specify preference for users (users who have not specified their preference)”. 4. The system prompts for confirmation from the PC chair to specify preference for PCs who have not specified a preference. 5. The PC chair submits his/her confirmation. 6. The system displays that the process is successful. 7. End | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:**  6a. If there are no papers submitted, the system displays that there are no papers submitted and goes back to the main page | |

|  |  |
| --- | --- |
| **Name:** Enable/Disable review submissions | **ID:** 23 |
| **Stakeholders and Goals:** Program Committee Chair – enable/disable review submissions | |
| **Description:** A program committee chair wants to enable or disable submission of reviews by the PCs. | |
| **Actors:** Program Committee Chairs | |
| **Trigger:** After logging in, program committee chairs gets to the page where he/she can enable or disable functionalities in the system. | |
| **Normal Flow:**   1. The Program Committee Chairs chooses the “Functionalities management” page. 2. The system lists all the functionalities that can be controlled by the PC chairs and prompts the PC chair on which he/she wishes to enable/disable. 3. The PC chair chooses “Enable/Disable review submissions”. 4. The system displays the status of submitting a review (Enabled or disabled). The system then prompts the PC chair to enable or disable. 5. The PC chair submits his decision. 6. The system prompts for reconfirmation 7. The PC chair submits confirmation 8. The system displays that submitting a review is now either disabled or enabled. 9. End | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:** None | |

|  |  |
| --- | --- |
| **Name:** Enable/Disable author response | **ID:** 24 |
| **Stakeholders and Goals:** Program Committee Chair – enable/disable author response | |
| **Description:** A program committee chair wants to enable or disable submission of response by authors on the reviews done on their own paper. | |
| **Actors:** Program Committee Chairs | |
| **Trigger:** After logging in, program committee chairs gets to the page where he/she can enable or disable functionalities in the system. | |
| **Normal Flow:**   1. The Program Committee Chairs chooses the “Functionalities management” page. 2. The system lists all the functionalities that can be controlled by the PC chairs and prompts the PC chair on which he/she wishes to enable/disable. 3. The PC chair chooses “Enable/Disable author response”. 4. The system displays the status of submitting a response (Enabled or disabled). The system then prompts the PC chair to enable or disable. 5. The PC chair submits his decision. 6. The system prompts for reconfirmation 7. The PC chair submits confirmation 8. The system displays that submitting a response is now either disabled or enabled. 9. End | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:** None | |

|  |  |
| --- | --- |
| **Name:** Enable/Disable review discussions | **ID:** 25 |
| **Stakeholders and Goals:** Program Committee Chair – enable/disable review discussions | |
| **Description:** A program committee chair wants to enable or disable discussion among reviewers on a particular paper. | |
| **Actors:** Program Committee Chairs | |
| **Trigger:** After logging in, program committee chairs gets to the page where he/she can enable or disable functionalities in the system. | |
| **Normal Flow:**   1. The Program Committee Chairs chooses the “Functionalities management” page. 2. The system lists all the functionalities that can be controlled by the PC chairs and prompts the PC chair on which he/she wishes to enable/disable. 3. The PC chair chooses “Enable/Disable review discussions”. 4. The system displays the status of review discussion among PCs (Enabled or disabled). The system then prompts the PC chair to enable or disable. 5. The PC chair submits his decision. 6. The system prompts for reconfirmation 7. The PC chair submits confirmation 8. The system displays that discussion of reviews is now either disabled or enabled. 9. End | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:** None | |

|  |  |
| --- | --- |
| **Name:** Group paper | **ID:** 26 |
| **Stakeholders and Goals:** Program Committee Chair – group paper based on topics | |
| **Description:** A program committee chair wants to manually group paper based on topics when the results of generating conference proceeding by the system is unsatisfactory. | |
| **Actors:** Program Committee Chairs | |
| **Trigger:** After logging in, program committee chairs gets to the page where he/she can start grouping papers based on topics manually. | |
| **Normal Flow:**   1. The Program Committee Chairs chooses the “Group papers manually” page. 2. The system lists all the papers submitted to the system and the group they are categorized in. The system then prompts which papers needs to be manually assigned to a new group. 3. The PC chairs chooses a paper and submits his/her choice. 4. The system then lists the groups currently available in the system and prompts which group the PC chairs wants to assign the paper to. 5. The PC chair submits his/her choice. 6. The system reconfirms PC chair’s decision. 7. The PC chair confirms his/her decision. 8. The system displays that the paper has been successfully assigned to the new group. | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:** None | |

|  |  |
| --- | --- |
| **Name:** Assign PC chairs | **ID:** 27 |
| **Stakeholders and Goals:** Admin – assign PC chairs | |
| **Description:** Admin wants to assign a new user with the role of “Program Committee Chair”. | |
| **Actors:** Admin | |
| **Trigger:** After logging in, administrator goes to the page where he/she can assign user the role of “Program Committee Chair”. | |
| **Normal Flow:**   1. The Admin chooses the “Assign PC chairs” page and brought to the page where he/she can assign new Program Committee Chairs. 2. The system lists down the users currently registered to the system. The system prompts the admin on which user needs to be assigned the new role. 3. The admin enters his/her choice. 4. The system reconfirms with the admin if he/she wants to assign this user with the role of “Program Committee Chair” 5. The admin confirms his decision 6. The system displays that the user has successfully been assigned the role of Program Committee Chair | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:** None | |

|  |  |
| --- | --- |
| **Name:** Create Conference | **ID:** 28 |
| **Stakeholders and Goals:** Admin – create new conference | |
| **Description:** Admin wants to create conferences. | |
| **Actors:** Admin | |
| **Trigger:** After logging in, administrator goes to the page where he/she can create a new conference. | |
| **Normal Flow:**   1. The Admin chooses the “Create new conference” page and brought to the page where he/she can create new conference. 2. The system prompts the admin for the details about the conference. 3. The admin inputs details about the new conference. 4. The system reconfirms the new details submitted by the admin 5. The admin confirms the submission. 6. The system displays that the conference has been successfully created. | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:** None | |

**Domain Model**

**Data Dictionary**

|  |  |
| --- | --- |
| **Class Name:** Author | **ID:** 1 |
| **Superclasses:** None | |
| **Attributes:**   * userID. Integer type. Assigned when account is created. * firstName. String type. * lastName. String type. * userName. String type. Can consist of numbers or alphabets. * email. String type. * password. String type * university. String type. * expertise. String type. Words or phrases describing the expertise of the user. | |
| **Methods:**   * modifyDetails()   Function to modify the attributes of the class.   * modifyPassword()   Function to allow modification of the password   * verifyPassword()   To be used when user log into the system.   * submitPapers()   Function to submit a paper.   * modifyPaperSubmission()   Function to modify attributes in the ResearchPaper class.   * sendResponse()   Function for user to leave a comment on their research paper after it has been reviewed. | |

|  |  |
| --- | --- |
| **Class Name:** ProgramCommittee | **ID:** 2 |
| **Superclasses:** Author | |
| **Attributes:**   * preferences. This will be an array of integers specifying the preference. 1 for “Yes”, 2 for “No”, 3 for “Maybe” and 4 for “Conflict of Interest”. The size of the array is based on the number of papers submitted. * paperAssigned. This will be an array of integers specifying the papers assigned to the ProgramCommittee. The size of this array depends on the number of papers per reviewers set by the Program Committee Chair.   \*Other attributes inherited from class Author. | |
| **Methods:**   * specifyPreferences()   Function to allow Program Committee to assign preference on all papers submitted.   * assignPapers()   Function to assign papers to ProgramCommitee.   * reviewPaper()   Function to allow reviewing of paper   * modifyReview()   Function to allow modification of review   * discussReview()   Function to allow discussion among reviewers. | |

|  |  |
| --- | --- |
| **Class Name:** ProgramCommitteeChair | **ID:** 3 |
| **Superclasses:** Author, ProgramCommittee | |
| **Attributes:**  \*Attributes inherited from ProgramCommittee and Author. | |
| **Methods:**   * assignProgramCommittee()   Function to allow assignment of Authors to the role of ProgramCommittee.   * monitorProgramCommittee()   Function to allow monitoring of the program committee.   * checkEvents()   Function to allow viewing of the latest events happening within the system.   * autoAssignPapers()   Function to allow automatic assignment of papers to ProgramCommittee   * autoSpecifyPreferemces()   Function to allow automatic assignment of preferences on papers for the ProgramCommittee.   * callAssignPapers()   Function to allow manual assignment of papers. Calls function “assignPapers” from the ProgramCommittee class.   * setReviewersNum()   Function to allow setting the global number of reviewers per paper.   * setPapersNum()   Function to allow setting the global number of papers per reviewer.   * manualSetPapersNum()   Function to allow manually setting the number of papers a reviewer receives.   * generateConfProc()   Function to allow the generation of conference proceeding. Taking the list of accepted papers and making a list.   * groupPaper()   Allow manual assignment of group for a paper.   * allowReview()   Function to enable/disable paper reviews   * approvePaper()   Function to accept or reject a paper.   * allowRespond()   Function to allow author response to their papers   * allowDiscussion()   Function to allow discussion among reviewers. | |

|  |  |
| --- | --- |
| **Class Name:** Administrator | **ID:** 4 |
| **Superclasses:** Author, ProgramCommittee, ProgramCommitteeChair | |
| **Attributes:**  \*Attributes inherited from Author, ProgramCommittee and ProgramCommitteeChair | |
| **Methods:**   * assignRoles()   Function to allow assignment of roles to different users. | |

|  |  |
| --- | --- |
| **Class Name:** ResearchPaper | **ID:** 5 |
| **Superclasses:** None. | |
| **Attributes:**   * paperID. Integer type. * constributedUsers. Array consisted of the author IDs to represent those who have contributed to the paper. * title. String type. * keywords. String type. Words that describe the research paper. * abstract. String type. Short paragraph providing a summary of the research paper. * assignedGroup. String type. Group that it has been assigned to. Used for generating conference proceeding. * discussion. String type. Discussion that is done among the reviewers. * approval. Boolean type. Integer to represent if the paper is approved or not. 1 for “Yes”, and 2 for “No”. * userResponse. String type. A short comment sent in by the user regarding the reviews done on his paper. | |
| **Methods:**   * modifyAttributes()   Function to modify attributes of the class.   * assignUsers()   Function to link to the authors based on the email address input by the user.   * assignGroup()   Function to assign groups to the paper. | |

|  |  |
| --- | --- |
| **Class Name:** PaperReview | **ID:** 6 |
| **Superclasses:** None | |
| **Attributes:**   * paperID. Integer type. * strengths. String type. Strengths of the paper. * weakness. String type. Weaknesses of the paper * comments. String type. Comments left by the reviewer. * suitability. Integer type. Is the paper suitable as a short paper. * pcRemarks. String type. Remarks left by the Program Committee * overall. Integer type. Overall evaluation of the paper. * reviewerConfidence. Integer type. * relevance. . Integer type. * originality. Integer type * significance. Integer type * presentation. Integer type. * technicalQuality. Integer type. * evaluation. Integer type. | |
| **Methods:**   * modifyAttributes()   Function to allow modification of attributes in the class. | |

|  |  |
| --- | --- |
| **Class Name:** Conference | **ID:** 7 |
| **Superclasses:** None | |
| **Attributes:**   * conferenceName. String type. * conferenceYear. Int type. Year that the conference is held on * conferenceDetails. String type. Short description on the conference. | |
| **Methods:**   * modifyAttributes()   Function to allow modification of attributes in the class. | |

**Appendix**

**Work diary – Entry 1**

**Siu Tung Lee**

15/08/2014 (Wk 1)

09:03 PM

Things I did: Looking for a group and then meet with Hnngh Wee. Summary the assignment questions, and try to understand the needs of the system.

End of view 12:31 PM

----------------------------------------------------------------------------------------------------------

17/08/2014 (Wk 1)

10:30 PM

Things I did: I have drawn down the step of the system that how to work in the assignment.

End of view 12:30 PM

----------------------------------------------------------------------------------------------------------

21/08/2014 (Wk 2)

11:15 AM

Things we did: Group discussion, plan with meeting with client at 11:30 AM and ask for the questions.

Meeting with client end at 12:25 PM.

Group meeting start at 12:30PM.

Things we discussed: We discussed about the information from client and start to work on it, and I showed the summary of the assignment and the steps.

Tasks to complete for next group meeting:

* Find out the complier that we can use in Windows.

End of meeting 12:30 PM

----------------------------------------------------------------------------------------------------------

24/08/2014 (Wk 2)

03:30 PM

Things I did: I am looking for the complier in the internet. And I found out the step of the installing complier

End of view 12:30 PM

----------------------------------------------------------------------------------------------------------

25/08/2014 (Wk 3)

2:00 PM

Things we did: Group meeting. We have looked on the C++ program design and I showed the complier installation. Giritharan Govindasamy showed his main page design.

What we discussed: We discussed about the implementation of the software from the clients requirements and specifications. Group leader wee ask me to start designing the use case model.

Task to be complete for the next meeting:

- Start designing the use case model.

End of meeting 4:30 PM

----------------------------------------------------------------------------------------------------------

27/08/2014 (Wk 3)

10:43 AM

Things we did: Preparing for meeting with client in lab. And we asked more information from client.

Tasks to complete for next group meeting:

- continue the use case models and install the software for doing use case model.

- I need to show the use case models to the group.

End of lab 11:46 AM

----------------------------------------------------------------------------------------------------------

28/08/2014 (Wk 3)

10:30 PM

Things I did: I have installed the software for use case models and continue the use case studies.

End of view 12:30 PM

----------------------------------------------------------------------------------------------------------

31/08/2014 (Wk 3)

10:30 PM

Things I did: I have drawn down the use case in ver1.

End of view 12:30 PM

----------------------------------------------------------------------------------------------------------

01/09/2014 (Wk 4)

10:30 PM

Things I did: I have showed the use case ver1 to group leader, but there have some problems. I need to do the correction

End of view 12:30 PM

----------------------------------------------------------------------------------------------------------

04/09/2014 (Wk 4)

3:20 PM

Things we did: Meeting start at 3:30 PM. The group checked out the use case ver2 and makes the correction. Our group leader Wee discussed with me and helps me on the Use Case Models. Then I made the use case ver3 and posted it on the sharing place.

What we discussed: Changes to SRS introduction. We discussed and worked on the Use case diagram and the design and class diagrams. I and Girtharan are discussing about the class diagrams and correct some of the class diagrams.

End of meeting 5:30 PM

----------------------------------------------------------------------------------------------------------

10/09/2014 (Wk 5)

10:21 PM

Things I did: I have posted my working diary on our group facebook page for SRS report.

End of view 10:22 PM

----------------------------------------------------------------------------------------------------------

**Work diary – Entry 2**

**Giritharan**

**11/08/2014 (Monday) (Week 1)**

1330 hrs

What we do: We had our first group meeting for CSCI222 Assignment 1. Group members were assigned to individual roles and a group leader was assigned.

Discussion: Discussed about the management system and listing down the questions to be asked to the client.

Task to be complete for the next meeting:

- Prepare Questions to ask the clients about specifications

Meeting Ended at 1440 hrs

1845 hrs

What I did: I read through the management system assignment 1 and gathered a list of questions to be asked in the next meeting.

End of task work 2015 hrs

**13/08/2014 (Wednesday) (Week1)**

1030 hrs

What we do: We met up with the client in lab and ask the list of questions which we came up. Manage to clear some of our doubts. Questions were mainly about the software requirements.

Discussion: After meeting with the client, we discussed about the answers regarding about the software requirements. Group Leader Wee told us to come out with a simple UML regarding about the understanding of the system

Meeting Ended at 1230 hrs

1910 hrs

What I did: Started drawing out a simple UML Diagram about the Management system.

End of task work 2200 hrs

**14/08/2014 (Thursday) (Week 1)**

1530 hrs

What we do: Group Leader Wee check our UML diagram to see our understanding of the system. He checked if we have any mistakes on the system structure. While checking he also combined all the other group members chart to see if it is the same. By doing this he can find out if we all heading to the same direction.

Discussion: We discussed about the clients questions and answers. And what are the new roles for each of our members to be done in the next meeting.

Task to be complete for the next meeting:

- To list down the non functional requirements from the system management assignment 1.

- Start doing designing in C++ for the software.

Meeting Ended at 1630 hrs

2130 hrs

What i did: Started designing the main page of the system management in C++. Listing down some of the non requirements for the system management.

End of task work 2300 hrs

**2108/2014 (Thursday) (Week 2)**

1130 hrs

What we do: Met with the client in the lab to ask more information for the requirements we discussed in the earlier group meeting

Client meeting Ended at 1225 hrs

1230 hrs

Discussion: We discussed in the group regarding the information we ask from the clients about the requirements and specifications.

Meeting Ended at 1240 hrs

Task to be complete for the next meeting:

- Listing more non-requirements for the system

- Continue designing and coding for the system main menu.

2230 hrs

What i did: Continue to find the non functional requirements for the system and did designing and coding for the system.

End of task work 2345 hrs

**23/08/2014 (Friday) (Week 2)**

1930 hrs

What i did: Continue doing the coding for the design phase. Listed down a few non functional requirements to be presented to the group leader in the next meeting.

End of task work 2145 hrs

**25/08/2014 (Monday) (Week 3)**

1400 hrs

What we do: Group Leader Wee check my C++ program design and told me to source new ways to improve the design. I presented the program to the rest of the group members. I have also show few of the non-functional requirements to the group leader.

Discussion: We discussed about the implementation of the software from the clients requirements and specifications. Group leader wee ask me to start designing the domain model

Task to be complete for the next meeting:

- Start designing the Domain model

- To continue and improve the designing of the software

Meeting Ended at 1630 hrs

1500 hrs

End of task work 1515 hrs

**27/08/2014 (Wednesday) (Week 3)**

1400 hrs

What we do: Met up with the client in the lab to reconfirm the requirements.

Discussion: We discussed if we have missed any information regarding the requirements.

Task to be complete for the next meeting:

- Continue designing the Domain model

- To continue and improve the designing of the software

Meeting Ended at 1630 hrs

End of task work 1515 hrs

**29/08/2014 (Friday) (Week 3)**

2030 hrs

What i did: Pause the designing of the software in c++ and continued doing the domain model using agroUML software windows. Using the requirements to design the classes.

End of task work 2200 hrs

**04/09/2014 (Thursday) (Week 4)**

1520 hrs

What we do: Presented with my group the Domain model. Had a few errors, Our group leader Wee discussed with me and help me improve the model by adding and correcting the mistakes i done.

Discussion: Editing the old version of SRS introduction to the latest version. We also discussed about the USE CASE diagram the other team members did. We draw the USE CASE diagram in ArgoUml software.

Meeting Ended at 1730 hrs

End of task work 1735 hrs

**10/09/2014 (Wednesday) (Week 5)**

1100 hrs

What we do: Met with the group to do formatting of SRS and others. Handed my Diary to Group Leader Wee.

Meeting Ended at 1230 hrs

End of task work 1235hrs

**Work diary – Entry 3**

**Toby**

31/7/2014 (Week 1)

5:30 pm

Things I did:

I have created a Facebook Group which added every member into the group. We exchanged our contact details as well.

End time 5:50pm

9/8/2014 (Week 1)

9:00 pm

Things I did:

I have gone through the assignment sheet. I dropped notes which helped me to understand the requirement. I have figured out the interview report doesn’t help a lot with the task.

End time 10:30 pm

12/8/2014 (Week 1)

10:45 pm

Things I did:

I have gone through the lecture notes about requirements and I started making up questions which I have to ask the client for the program. I realized there were many unclear points of the interview report. I came up with one to two big issues.

End time 12:30am

13/8/2014 (Week 1)

8:30 pm

Things I did:

Our leader has come up with an Attendance List and a General Rules. This would keep control to the group attendance and task quality. I have gone through them til I understand our responsibility.

End time 8:45 pm

14/8/2014 (Week 1)

3:30 pm

Things we did:

Group meeting

Task completed: discussion of the conference planning, questions for the client.

We had a group meeting about the questions to the client. I didn’t have a good understanding with the system process. My group members gave me some advices which helped me out with the processing chart.

End time 4:30 pm

21/8/2014 (Week 2)

11:26 am

Things we did:

Group meeting

Task completed: met with our client, collect information, assign task to every member.

We had a meeting with our client, now the specification is getting clearer.

End time 12:30 pm

24/8/2014 (Week 2)

10:00 pm

Things I did:

I read through the sample on the e learning, realized it and come out with the business case. After I have done, I uploaded it.

End time 12:00 am

26/8/2014 (Week 2)

11:00 pm

Things I did:

I got assigned to work with Damon on the Risk and counter measures section. After I have done, I uploaded it.

End time 12:30 am

29/8/2014 (Week 2)

9:45 pm

Things I did:

I got assigned to work with Damon on the introduction section. I started working on the purpose and overview parts. We talked and come out with some opinions.

End time 11:30 pm

1/9/2014 (Week 3)

5:51 pm

Things I did:

Damon and I have combined our work to complete the introduction. We edited a bit of both of our work. We uploaded it after we have done.

End time 5:55 pm

4/9/2014 (Week 3)

10:30 am

Things I did:

I got assigned to work with [Giritharan](https://www.facebook.com/giritharan.govindasamy) on the class diagram. We met up in the library and gone through again the task process, tried to work out the general structure.

End time 12:00 pm

4/9/2014 (Week 3)

9:45 pm

Things I did:

I got assigned to do the submit paper and review side for the class diagram. Honestly, this task is a bit hard, I tried to go through each functions, see whether I could separate them to different classes by referring to their characteristic.

End time 12:00am

5/9/2014 (Week 3)

9:30 am

Things we did:

Group meeting

Task completed: use case completed, tasks combined, discussion on the class diagram.

We had a meeting again. We tried to combine what we had got at the moment. The use case has all most done. We are still working on the class diagram.

End time 1:00 pm

**Work Diary – Entry 4**

**Damon**

11/08/2014 (Wk 1)

1:30 PM

Things we did: First group meeting regarding assignment 1. Assigned group leader.

What was discussed: Review/explanation of assignment, look at tasks in detail. Viewed tasks, report and marking scheme.

Viewed program required to build (conference management system), viewed key features, and interview of staff/questions about conference in general, and about what they expect in the software they require. Viewed Appendix A (sample review form).

Tasks to complete for next group meeting:

- General chart for process for conference planning

- Prepare questions for lab to ask clients for specification

- View pre-existing software and provide quick summary of what's provided for next meeting.

End of meeting 2:40 PM

3:00 PM

Things I did: Started planning chart task for next meeting in word document "Conference plan chart.docx".

Stopped due to lecture in 10 minutes.

End of task work 3:22

8:56 PM

Things I did: Continued planning chart task.

Finished 9:10 PM

Started viewing added websites and pre-existing software in assignment 1 handout.

Preparing quick summary on each in "Website Eval.txt".

Finished summary.

End of task work 9:30 PM

12/08/2014 (Wk 1)

10:06 PM

Things I did: Started preparing questions to ask for specifications for clients in “Questions.docx".

Finished my preparing of questions, am unsure what to ask for software requirements without knowing specific tasks for application. Will see at meeting and lab (asking clients) tomorrow.

End of task work 10:19 PM

13/08/2014 (Wk 1)

10:30 AM

Things we did: In lab, preparing questions to ask. Not brilliant questions provided by me.

What was discussed: Our group came up with a basic idea in what to ask in terms of requirements and why each are in there, and we have a list (mostly done by Shein) to ask for more detail about the assignment. I will record the question answers (not for the group) in the file "Question answers.docx". We have also come up with our next meeting time for tomorrow, which is at 3:30 PM

Had questions answered, client interview was finished 12:02 PM

End of lab work 12:10 PM

14/08/2014 (Wk 1)

3:30 PM

Things we did: Group meeting, created a schedule up to Assignment 1 finish date.

Tasks completed for meeting:

-General chart for conference planning

-Client questions.

-Pre -existing software and quick summary.

What we discussed: Discussed graphs and work we had to complete from last meeting. Discussed SR sheet with questions to ask for next lab.

Tasks to complete for next group meeting:

-Have basic outline on web task for weekend

-More questions to ask if needed from read through outline.

End of meeting 4:30 PM

21/08/2014 (Wk 2)

11:15 AM

Things we did: Group discussion, plan with meeting with client at 11:30 AM.

Meeting with client end at 12:15 PM.

Group meeting start at 12:17 PM.

Things we did: Starting to work on report business case.

Things we discussed: Allocated tasks, mention of basic design and major issue with one section on assignment.

Tasks to complete for next group meeting:

-submit command line - operations, command lines and locations (how the submit command line works)

End of meeting 12:30 PM

22/08/2014 (Wk 2)

11:50 PM

Things I did: Started work with submit line, finding out about how it works.

Work halted due to phone call about father's case from army. Issue erupted.

Unable to complete work, inaccurate work as guideline for change for future work.

Abrupt end of task work 12:20 PM.

23/08/2014 (Wk 2)

2:01 PM

Things I did: Continued task work for submit line. It is 2:28 PM and I cannot find much info about submitting files.

3:04 PM I have found two websites which seems to explain it for the project, using a website base which I can rig:

http://www.codeproject.com/Articles/2507/Basics-of-uploading-files-to-and-downloading-from

http://msdn.microsoft.com/en-us/library/windows/desktop/aa384180%28v=vs.85%29.aspx

End of task 3:14 PM 23/08/2014

25/08/2014 (Wk 2)

2:00 PM

Things we did: Group meeting. We had a look at the design of the program and the starting designs of each page. We are using command line so the design might be difficult. We completed the task business case and discussed it in this meeting as well.

What we discussed: Talked about submit issue, and possibility that we might have to use ftp protocol commands. We also discussed other options. We discussed changes to the SRS and about the risk assessment. Task assigned: complete risk and contingency plan with Toby.

We formed a list of possible questions and decided to review the requirements in the client meeting on Wednesday.

End of meeting 3:00 PM

26/08/2014 (Wk 2)

12:56 PM

Things I did: Started task set, about risk and contingency plan. Set time as I am due for a lab exercise, I have set a time limit to be half an hour. Cannot think of other risks for the time being. I have uploaded it to the group facebook page for any other risks they can think of.

End of task 1:15 PM

27/08/2014 (Wk 2)

10:43 AM

Things we did: Preparing for meeting with client in lab. The plan is to be seen and ask questions before 12:30. Finished confirming requirements, have implemented changes.

Tasks to complete for next group meeting:

-Toby and I are to complete the SRS introduction (mainly purpose, scope and overview).

End of lab 11:46 AM

31/08/2014 (Wk 3)

11:23 PM

Things I did: Starting the task of the SRS introduction scope. My plan is to have this finished within the hour.

Scope completed with little interruption.

End of task 12:32 PM

1/09/2014 (Wk 3)

5:44 PM

Things I did: Working on introduction for SRS continued. I have joined with Toby to complete the task.

Task completed and uploaded to the SRS.

End of task 5:55 PM

4/09/2014 (Wk 3)

3:20 PM

Things we did: Meeting start at 3:30 PM. Original plan was to help a fellow member correct and complete the use case and description. Unable to get in contact to continue the work with the group member.

What we discussed: Changes to SRS introduction for final SRS. We discussed and worked on the Use case diagram and the design and class diagram for the assignment. We also worked on a plan of the use case before we drew up the use case.

End of meeting 5:25 PM

8/09/2014 (Wk 4)

12:03 PM

Things I did: Reviewed the final versions of the assignment we have to hand in. Not sure how long this is going to take. There is nothing I can think of to update, so I will leave it for now.

End of view 1:31 PM

8/09/2014

8:03 PM

Things I did: I will hand in the work diary to my group facebook page.

End of task 8:05 PM

9/09/2014

8:18 PM

Things I did: I had to re-format my work diary for submission according to project leader.

End of task 8:19 PM