

A dark blue vertical bar runs down the left side of the page. A blue arrow points to the right from this bar, containing the date.

9/4/2015

# SI/ISM354 Group Project Documentation: Analysis & Design

Several thin, dark blue wavy lines originate from the bottom of the vertical bar and curve upwards and to the right.

Group number: 1D

Tiaan du Toit 17038456

Aron Frost 17071186

Marco Lourenco 17127211

Lujané Sieberhagen 17640776

## Contents

Version Control.....	1
RACI Chart .....	1
Executive Summary .....	2
Overview .....	2
Background .....	2
Objectives .....	2
Requirements .....	2
Proposed Strategy .....	2
Next Steps .....	3
Scope.....	3
Included in Scope.....	3
Excluded from Scope.....	3
Constraints .....	3
Risk Analysis .....	4
<b>Technological Risks</b> .....	4
<b>Skills Risks</b> .....	4
<b>Requirements Risks</b> .....	4
Timetable .....	4
Business Use Case .....	5
Business Use-Case Diagram .....	5
Business Use-Case Descriptions.....	6
Business Use Case: Booking for conference .....	6
Business Use-Case: Submission .....	8
Call and response table: Form Validity .....	8
Business Use Case: Registration.....	10
Business Use Case: Reviewing.....	11
Call and Response Table: Paper Recommendations.....	11
Business Use Case: Editing Website .....	13
Actors.....	14
Workers.....	14
Business Actors.....	14
Other Systems .....	15
Role Map.....	15
User Requirements.....	16

System Use-Case Diagrams .....	16
Systems Use-Case Descriptions.....	19
System Use Case: Register .....	19
System Use Case: View Conference Website .....	20
System Use Case: Book for Conference .....	20
System Use Case: Cancel Booking.....	21
System Use Case: Sign In .....	21
System Use Case: View Profile .....	22
System Use Case: Delete Profile .....	23
System Use Case: Submit Paper .....	23
System Use Case: View status of Paper .....	24
System Use Case: Update Paper .....	24
System Use Case: View Assigned Papers .....	25
System Use Case: Submit Review .....	26
System Use Case: Update Review .....	27
System Use Case: Assign Papers .....	27
System Use Case: View ALL Papers.....	28
System Use Case: View Reviews .....	28
System Use Case: Update Conference Website.....	29
System Use Case: View Registered Members.....	30
System Use Case: Assign Roles to Registered Members .....	30
System Use Case: Accept/Reject Papers .....	32
System Use Case: Manage Conference .....	33
State-Machine Diagrams.....	34
State-Machine Diagram: Papers .....	34
.....	34
State-Machine Diagram: Reviewers .....	35
.....	35
State-Machine Diagram: Users .....	36
Non-Functional Requirements .....	37
Performance Requirements.....	37
Usability Requirements.....	37
Security Requirements .....	37
Volume and Storage Requirements .....	37
Configuration Requirements .....	37

Compatibility Requirements .....	37
Reliability Requirements .....	37
Backup/Recovery Requirements.....	38
Training Requirements.....	38
Structural Model .....	39
Class Diagrams: Entity Classes .....	39
Package: People and Organizations .....	39
Package: Products and Services.....	40
Package: Events/Transactions.....	40
Specialization/Generalization .....	41
.....	41
Whole/Part Relationships.....	41
Associations .....	42
Test Plan.....	43
Implementation Plan .....	43
Training.....	43
Rollout.....	43
Sign-Off .....	43
Glossary .....	44
Reference List .....	45

## **Version Control**

### **RACI Chart**

\* - Authorize: This person has the final say on any changes made to the document.

R – Responsible: This person is responsible for making this document.

A – Accountable: This person is held accountable for the accuracy of this document.

S – Supports: This person provides support with the creation of this document.

C – Consulted: This person provides input. (Interviews etc.)

I – Informed: This person must be informed of any changes made to the document.

Name	Position	*	R	A	S	C	I
Tiaan du Toit	Group Member	X	X	X			X
Aron Frost	Group Member	X	X	X			X
Marco Lourenco	Group Member	X	X	X			X
Lujané Sieberhagen	Group Member	X	X	X			X
Howard Podeswa	Consultant				X		
James Johnson	Chairperson					X	

## **Executive Summary**

### **Overview**

This project is a conference management system for the researched conference held by the Stellenbosch Institute for Informatics and Software Management.

### **Background**

The program is being developed for the Stellenbosch Institute for Informatics and Software Management. This institute is a local academic computer society. In 2016 they will be hosting there first mayor conference.

This is an annual international academic conference about computer science. The conference covers a wide range of topic, thus resulting in it being a mayor event. It is a great opportunity for the institute to have received the opportunity to host this mayor event.

In order for the institute to run this event successfully, they require a multifunctional conference management system.

### **Objectives**

The institute cannot find a single program that contains all of the required features. This means that the mayor objective of this program is that it consists of all the required features and runs successfully.

### **Requirements**

The system has to be able to consist of four mayor abilities:

1. It has to have a full functional website. This website will be available for use by the general public. All of the conference information will be displayed on the site
2. All papers can be submitted on the website. The reviewing and acceptance or rejection of the papers will also take place on the system.
3. Registration of all participating parties and buying of tickets by potential delegates will also take place on the website. Each registered user has limited access to the system, depending on his/her role.
4. The chairperson has access to the back end of the system and can edit the website in any desired way.

### **Proposed Strategy**

In the discovery phase, the business analysts analyse all of the use cases. Next the design of the program will take place.

A soon as the designs have been accepted the coding will take place. The basic requirement will be coded first. After that has been completed, if possible, additional features will be added to the program.

## **Next Steps**

- **Action:** The software will be developed
- **Responsibility:** Tiaan du Toit  
Aron Frost  
Marco Lourenco  
Lujané Sieberhagen
- **Expected Date:** 5 October 2015

## **Scope**

### **Included in Scope**

The system will provide a developed functional conference management system. The system will be used to facilitate The Stellenbosch Institute for Informatics and Software Management conference in Stellenbosch in 2016. The conference system has three main components. It has the ability to provide a fully featured conference website for the general public. It has the ability to add the submission of academic papers, the review of these papers and the tasks which are associated with reviewing papers, the acceptance or rejection made by the Chairperson after the committee's recommendation. The system when performing the submission of papers and reviewing of papers has a "key words" matching ability to enable reviewers to read papers they would prefer. The final ability which the system enables is to provide the registration and booking for potential delegates and all relevant parties.

### **Excluded from Scope**

The system does not check if papers are submitted in the correct format, this is handled by members of the committee. The system does not track the status of papers automatically, it needs to be told by either the reviewer, member of committee or Chairperson. The system does not track conflicts between a paper being submitted and then reviewed but may be added if system development is ahead of schedule. The system will not track the payments for the bookings of the conference although this may be added if the project is ahead of schedule.

### **Constraints**

1. All members will be the main liaisons for the project.
2. Final approval is the 5<sup>th</sup> of October

## **Risk Analysis**

Strategies for dealing with risk:

- **Avoid:** Action taken to eliminate the risk
- **Mitigate:** Action taken to reduce damage of the risk if it occurs
- **Transfer:** Passing the risk to another entity
- **Accept:** No action taking to prevent the risk, and accept the consequences

### **Technological Risks**

**Risk:** Ensuring compatibility between internet browsers. Implementing the python code sufficiently in order to provide all interface and functionality desired.

**Likelihood:** Medium

**Cost:** Project delays

**Strategy:** Mitigate. Learn the correct coding needed in order to build a system with high compatibility and functionality.

### **Skills Risks**

**Risk:** Lack of competence in programming in the team. Python language and the Django web-development program is unfamiliar and requires more practise.

**Likelihood:** High

**Cost:** Project delays

**Strategy:** Avoid. Make sure the team learns the code before we start the project.

### **Requirements Risks**

**Risk:** Inconsistency between documentation and programming and development may occur because of team members being uncertain of whether the UML diagrams accurately represent what the stakeholders actually want.

**Likelihood:** Low

**Cost:** Faulty system

**Strategy:** Review our interview recordings and notes.

## **Timetable**

**Discovery:** Completed 17/08/2015

**Construction:** Plan to complete it by 05/10/2015

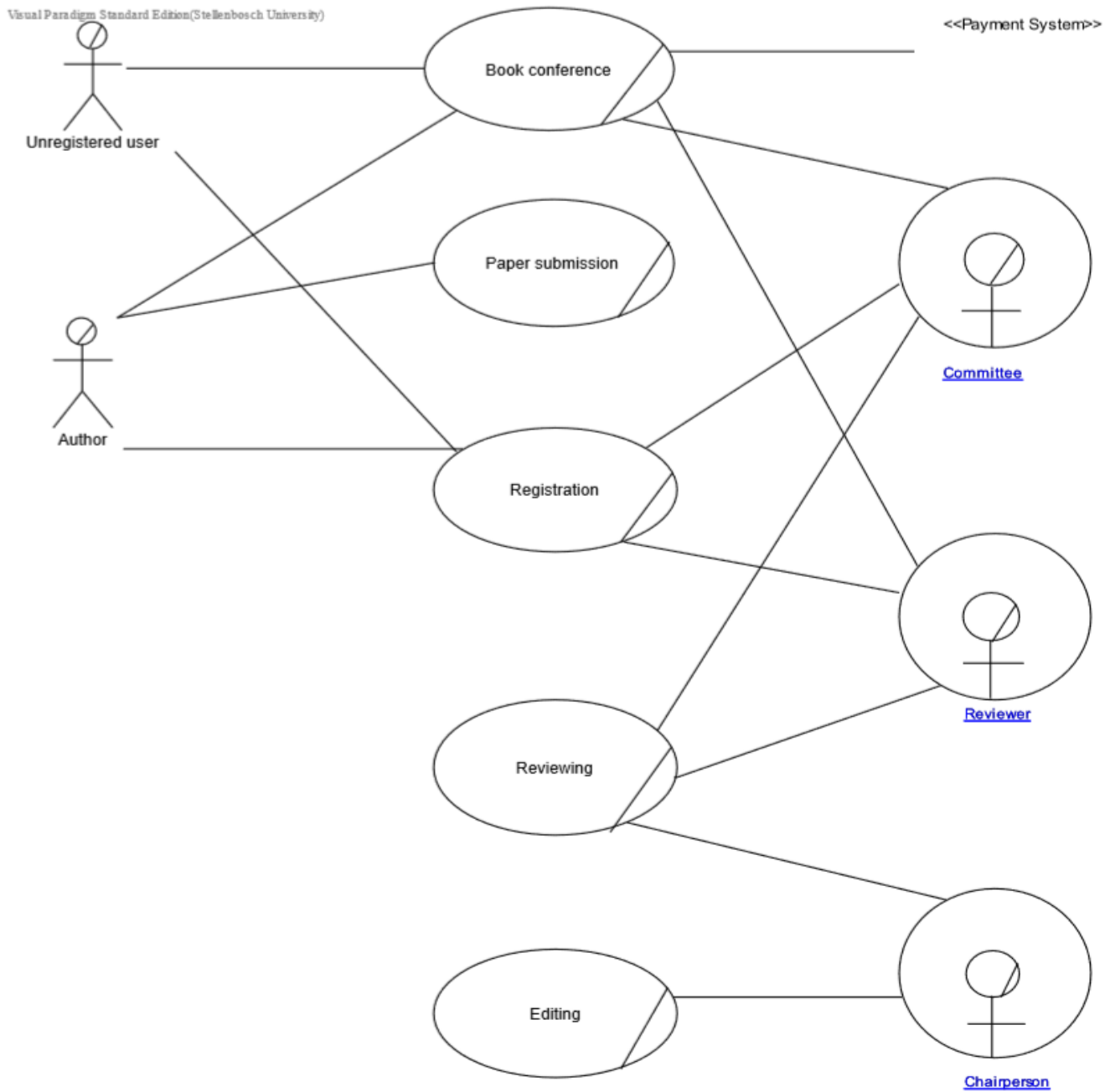
**Final V&V:** TBD

**Closeout:** TBD



## Business Use Case

### Business Use-Case Diagram



## **Business Use-Case Descriptions**

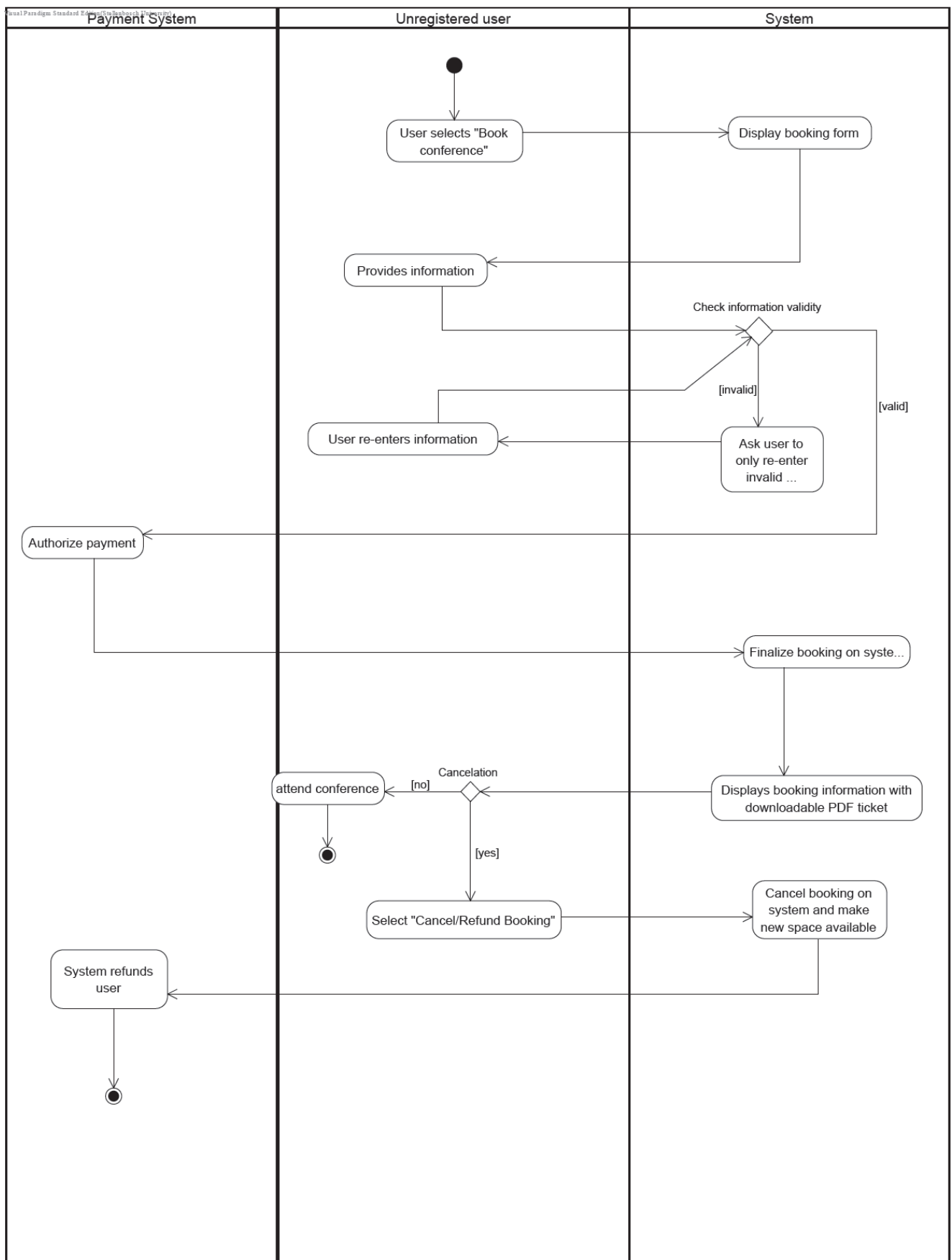
### **Business Use Case: Booking for conference**

**Pre-conditions:** Potential delegate enters the site to make booking.

**Post-conditions on success:** The system will print out a ticket for the delegate once payment has occurred.

**Flow:**

1. The potential delegate selects the book conference link
2. The system displays the booking form.
3. The potential delegate enters in his or her information.
4. The system confirms all fields are correct.
- 4.1 The Information is invalid and potential delegate has to re-enter the fields.
5. A third-party application will authorize the payment.
6. The system finalizes the booking by sending a ticket.
7. The delegate may cancel.
- 7.1 The system removes the delegate from the conference and provides refund.

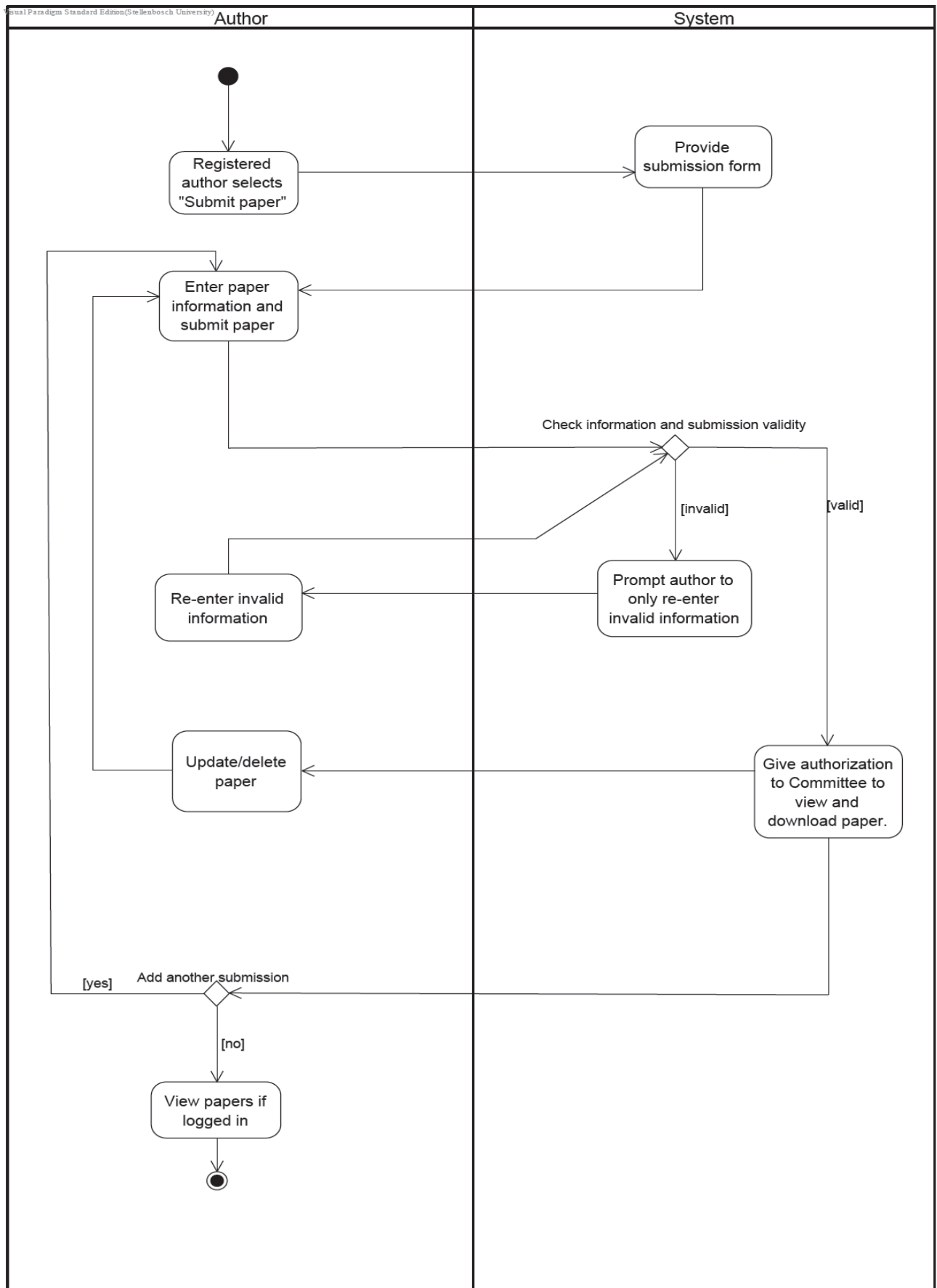
**Activity Diagram:**

**Business Use-Case: Submission****Pre-condition:** A User has registered as an Author.**Post-condition:** The Author's Paper has been submitted for review.**Flow:**

1. A registered Author selects "Submit paper" on their profile.
2. The Author selects the upload option and attaches their paper.
3. The Author enters data in all fields and is prompted to complete all fields before submitting.
4. The Author is prompted to re-enter only fields that are incorrect if applicable.
5. The Author is sent back to the submission screen with the option of uploading another paper.
6. The Author can either submit another paper or they are sent to their profile which shows papers already uploaded.
7. The paper is sent to the Committee so that they can view and download it.
8. The Author will be notified to update the paper if there is a problem with submission.
9. If the Author is logged on, they can view their submitted papers and paper status.
10. When the deadline for submissions is reached, an Author can no longer submit papers and submission option is unavailable, papers are under review.

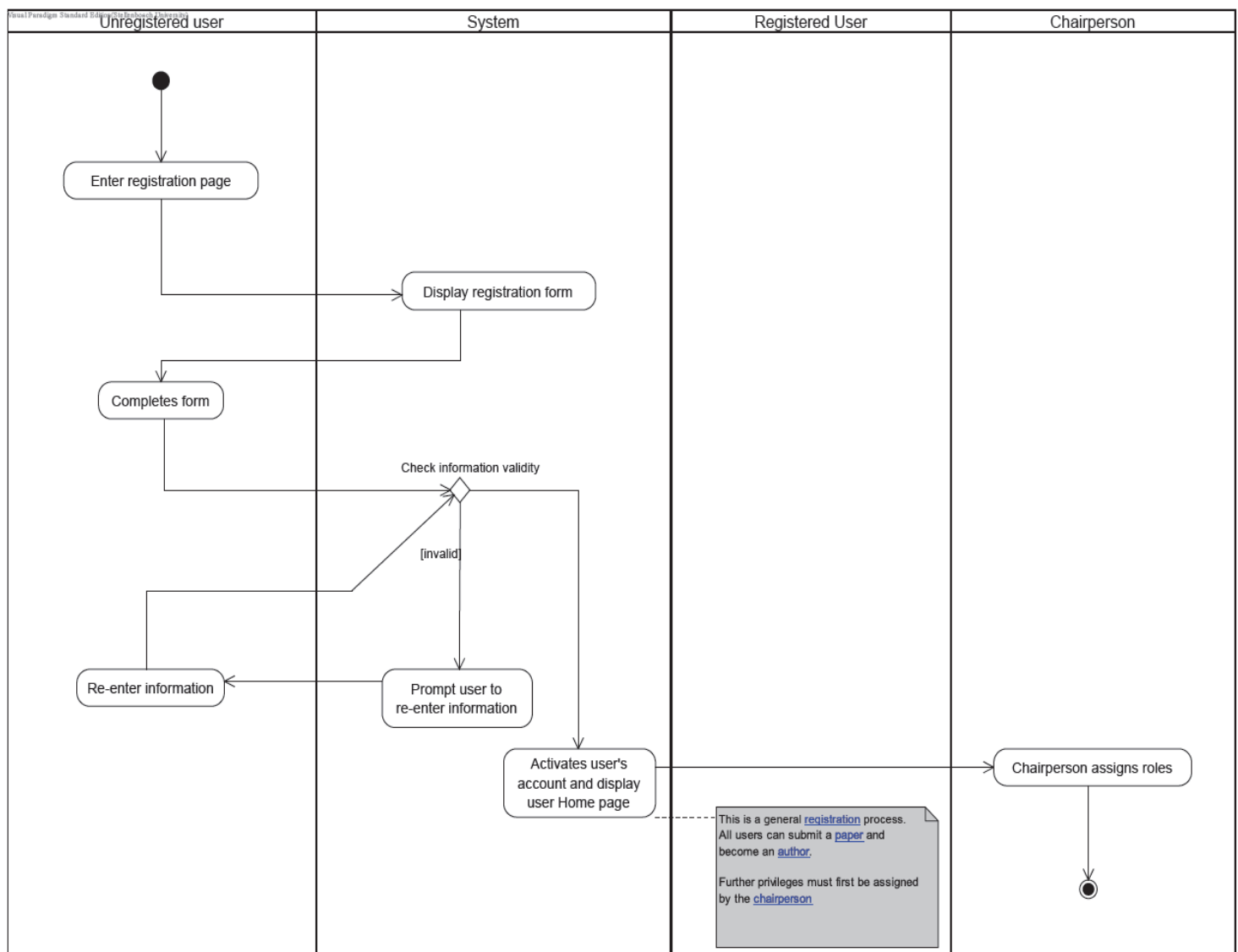
**Call and response table: Form Validity**

Condition	Response
Information entered incorrectly	Reject form and prompts user to re-enter info
Information entered correctly	System accepts information

**Activity Diagram**

**Business Use Case: Registration****Pre-condition:** An unregistered user has visited the website and wishes to create an account.**Post-condition on success:** The user registers successfully.**Flow:**

1. The unregistered user enters the registration page.
2. The system displays the registration form.
3. The user then completes the form.
4. The system checks if all the information added is valid. If not it prompt the user to re-enter the invalid information.
5. If all information is correct, the system creates the user's account.
6. The chairperson can now assign roles to registered users.

**Activity Diagram**

**Business Use Case: Reviewing**

**Pre-condition:** The papers have been submitted before the submission due date.

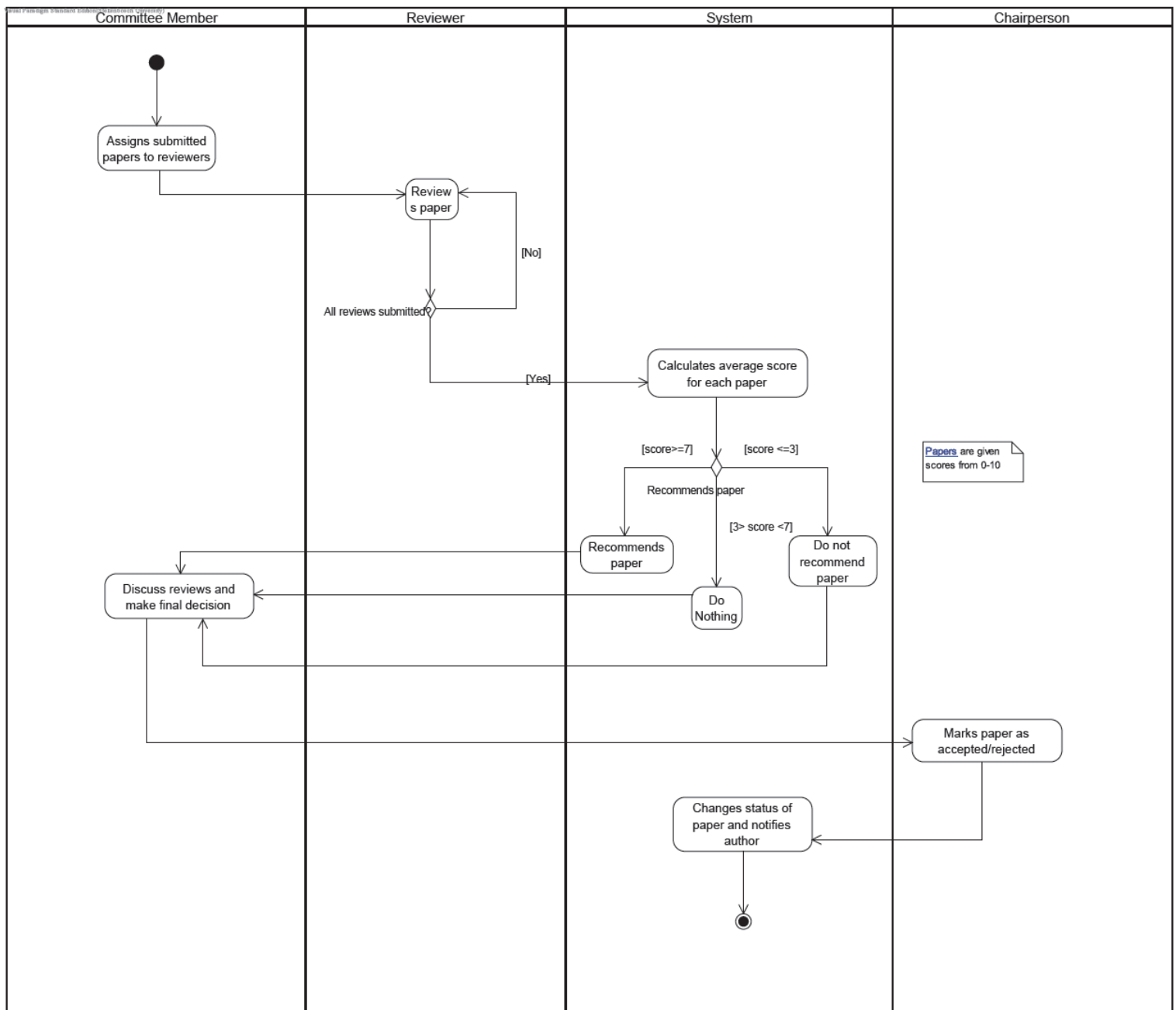
**Post-condition:** The accepted and rejected papers authors can view the papers status on their personal profile on the website. One of the authors of each accepted paper will have to register for the conference.

**Flow:**

1. The committee members assign reviewers to all of the papers.
2. The reviewer reviews the paper and submits the review.
3. All of the reviews have not been submitted.
  - a) Review the rest of the papers and submit them.
4. All of the reviews have been submitted.
5. The system calculates the average score for each paper.
6. The paper gets an average score of greater or equal to 7.
  - a) The system recommends the paper.
7. The paper gets a score of smaller or equal to 3.
  - a) The system does not recommend the paper.
8. The paper gets an average score between 3 and 7.
  - a) The system does nothing.
9. The committee members will then discuss the reviews and make a final decision on acceptance or rejection of the paper.
10. The chairperson will mark the paper as accepted or rejected on the system.
11. The system changes the status of the paper and notifies the author of the acceptance or rejection of his/her paper.

**Call and Response Table: Paper Recommendations**

<u>Condition</u>	<u>Response</u>
Avg. Score $\leq 3$	Do not recommend paper
$3 < \text{Avg. Score} < 7$	Do Nothing
Avg. Score $\geq 7$	Recommend paper

**Activity Diagram:**



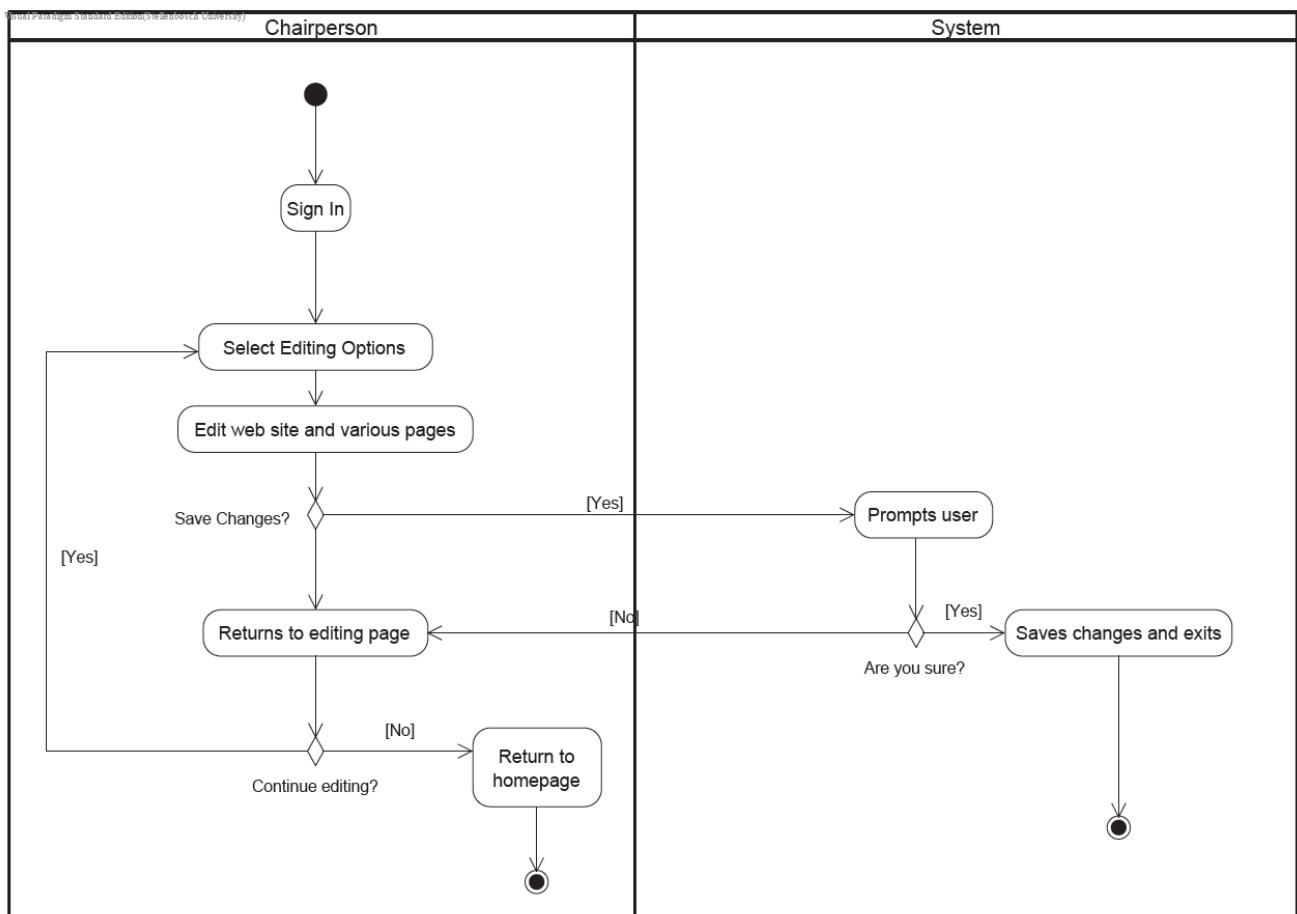
**Business Use Case: Editing Website**

**Pre-condition:** The chairperson must create various front-end web pages which provide information about the conference.

**Post-condition on success:** Various front-end web pages

**Flow:**

1. The chairperson signs in to his account.
2. The chairperson the selects editing options.
3. He then uses the tools provided by the system to edit the various pages.
4. The chairperson the saves his changes.
5. The chairperson can accept his changes and then returns to his profile page, or discard them and continue editing.



## **Actors**

### **Workers**

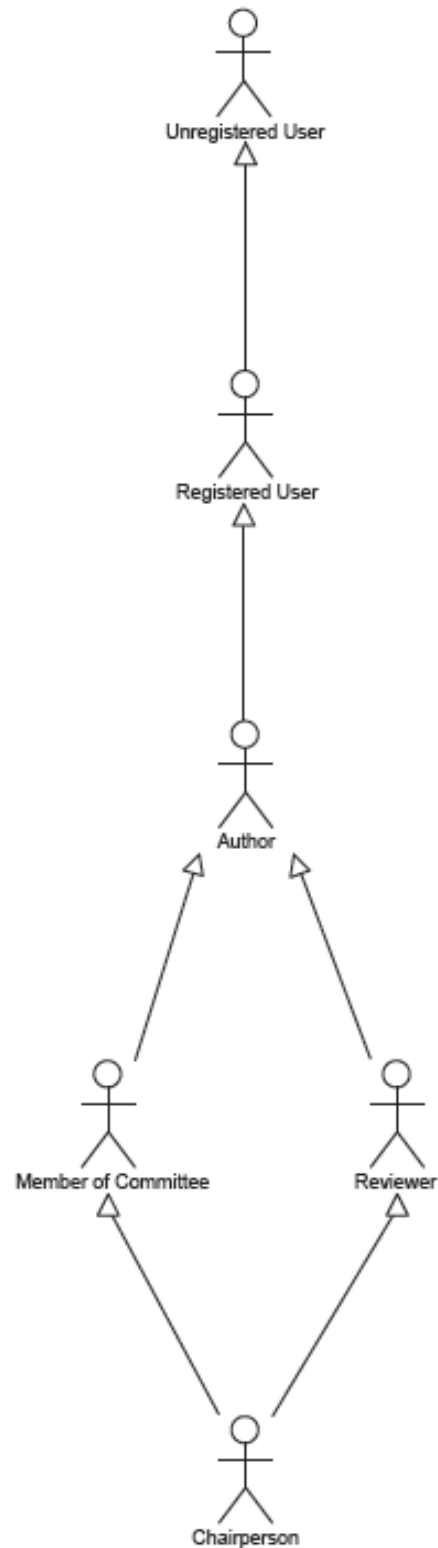
<b>DEPARTMENT/POSITION</b>	<b>GENERAL IMPACT ON PROJECT</b>
CHAIRPERSON	Head of the committee and convener of the conference. Will use IT to create various websites for the conference, as well as assigning roles to committee members and reviewers. Will also use the system to mark papers as accepted and rejected.
COMMITTEE MEMBER	A committee that discusses papers and their reviews. They will assist the chairperson in organizing the conference. He will use system to assign papers to reviewers, and make sure all reviewers have submitted all required reviews. Current manual processes of debating which papers to accept will remain in place.
REVIEWER	Is assigned papers to review. A score out of 10 and comments are required for each paper assigned to each reviewer. Will use IT to check assigned papers, download them and submit reviews.

### **Business Actors**

<b>ACTOR</b>	<b>GENERAL IMPACT ON PROJECT</b>
AUTHOR	An academic who is interested in attending the conference as a speaker. Will use IT to submit a paper to the conference committee and update submission if required.
DELEGATE	Attendee of the conference. Will use IT to view the conference website and program as well as booking tickets.
REGISTERED USER	A user who has the potential to become either an author, reviewer or committee member. Will use IT to check which roles have been assigned to them.
UNREGISTERED USER	Potential delegate visiting the website. They are interested in seeing information about the conference such as ticket prices and the program.

**Other Systems**

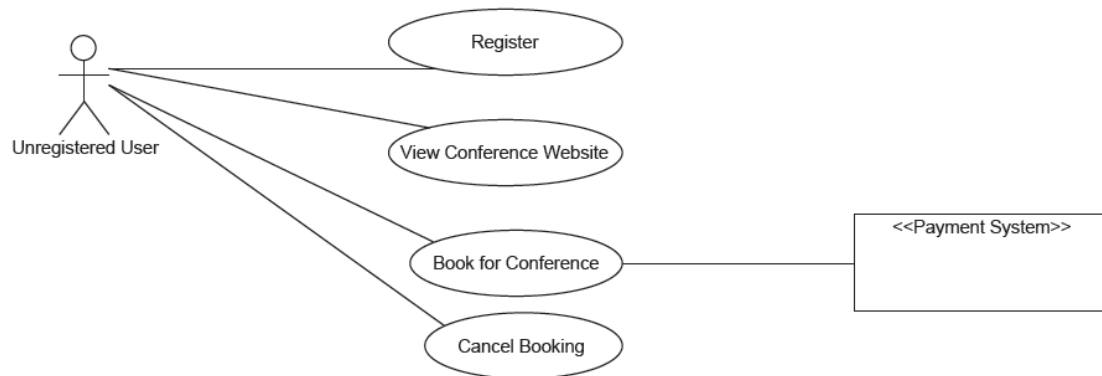
SYSTEM	GENERAL IMPACT ON SYSTEM
PAYMENT SYSTEM	Third-party application that is responsible for the transaction of tickets.

**Role Map**

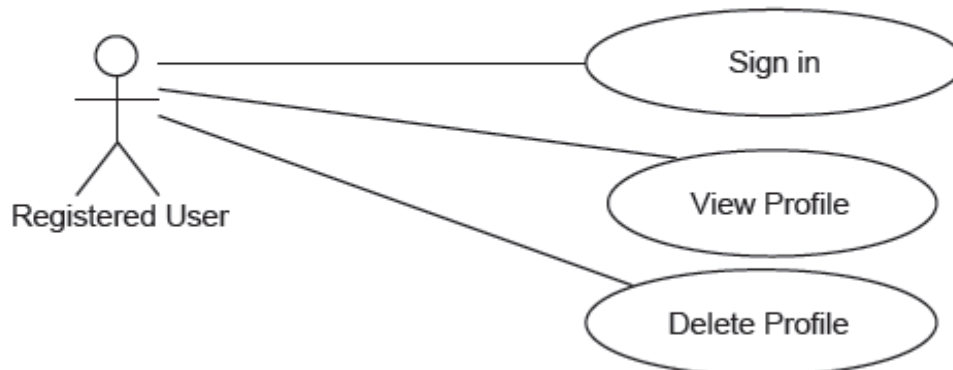
## User Requirements

### System Use-Case Diagrams

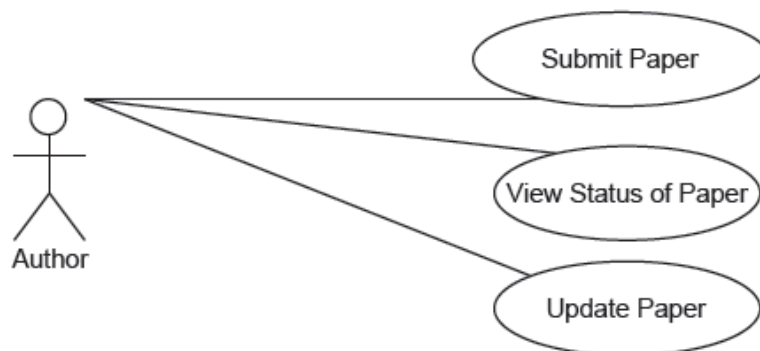
Visual Paradigm Standard Edition (Stellenbosch University)  
Non-Registered  
Delegate System Use



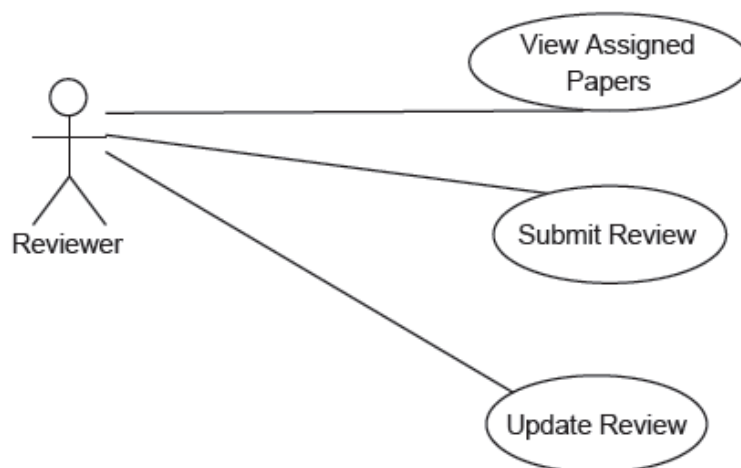
Delegate System  
Use Case



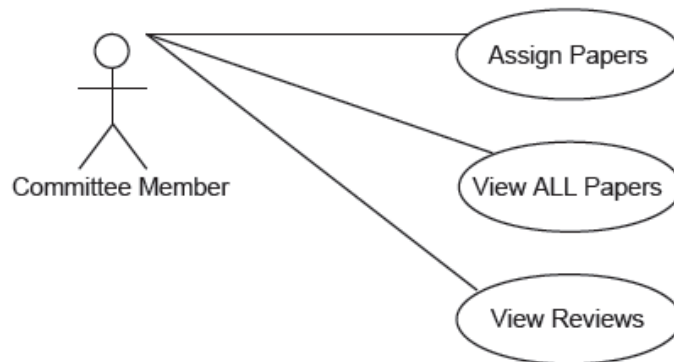
Author System  
Use Case



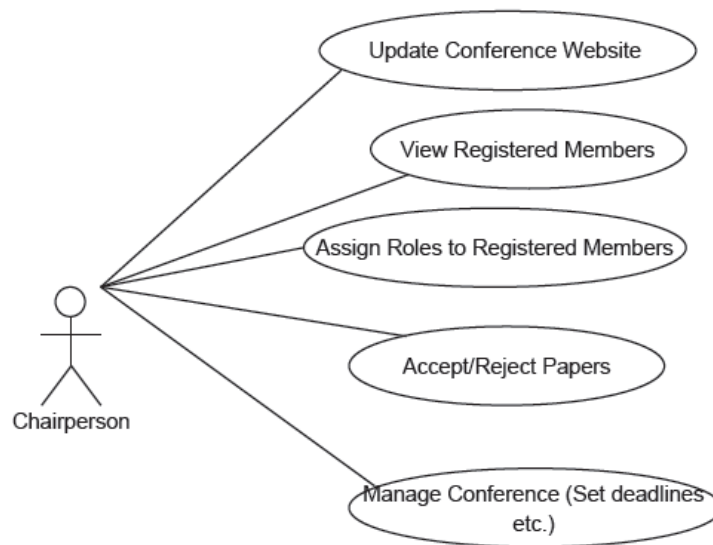
Reviewer  
System Use



Member of Committee  
System Use Case



Chairperson  
System Use Case



## Systems Use-Case Descriptions

### System Use Case: Register

**User story:** As a unregistered Delegate, I want to register, so that I can have a profile.

#### **Basic Flow**

1. User selects Register Now.
2. Completes registration form.
  - 3.1 User presses submit
  - 3.2 Email is sent to confirm registration if successful.
  - 3.3 User is prompted to fill in blank or incorrect fields if unsuccessful.

#### **Alternative Flow**

Trigger: User resets the form

Divergence Point: 2

#### **Interface screenshot**

Please complete this Registration Form

Name	e.g. Bobby	<input type="checkbox"/>
Surname	e.g. Brown	<input type="checkbox"/>
ID number	e.g. 1204049094	<input type="checkbox"/>
Email Address	e.g. bobbybrown@sun.ac.za	<input type="checkbox"/>
Institute	e.g. University of Stellenbosch	<input type="checkbox"/>
Password	••••••••	<input type="checkbox"/>
Dietary Requirements	e.g. Halaal	<input type="checkbox"/>

Submit Reset

### **System Use Case: View Conference Website**

**User story:** As an unregistered Delegate, I want to view conference website, so that I can see conference details.

#### **Basic Flow**

1. Unregistered user visits home page of conference
2. Unregistered user views various pages including the conference program.

### **System Use Case: Book for Conference**

**User story:** As an unregistered Delegate, I want to book for the conference, so that I can attend.

#### **Basic Flow**

1. User selects Purchase Ticket.
2. User enters details.
3. User is prompted to pay via credit card and price is shown.
4. User receives notification that payment is successful, and tickets are emailed to user.
5. User needs to edit details or retry payment if unsuccessful.

#### **Alternative Flow**

Trigger: User decides to pay later, and saves details before exiting.

Divergence Point: 3

Convergence Point: User decides to make payment closer to the time. Returns to point 3 when site is accessed again.



## Interface screenshot

Please fill this out to Purchase Ticket

Name: e.g. Bobby

Email Address: e.g. bobby@sun.ac.za

Amount: [Dropdown menu]

Payment Type: Visa

Total Price: R200.00

Buttons: Make Payment, Save and Continue Later

Callout 1 (pointing to Total Price): System calculates and displays total price of ticket/s ordered

Callout 2 (pointing to Amount dropdown): Select Quantity of tickets

### System Use Case: Cancel Booking

**User story:** As a Non-registered Delegate, I want to be able to cancel bookings, so that I can change my decision if necessary.

#### Basic Flow

1. Delegate clicks on Booking tab on home page
2. Delegate then clicks on Cancel booking
3. The Delegate is refunded and the tickets become re-available for purchase.

### System Use Case: Sign In

**User story:** As a registered user I want to be able to sign in to my profile

#### Basic Flow

1. User browses to Log In
2. User fills in details.
3. User presses Submit
4. User can try again if sign in details are incorrect.

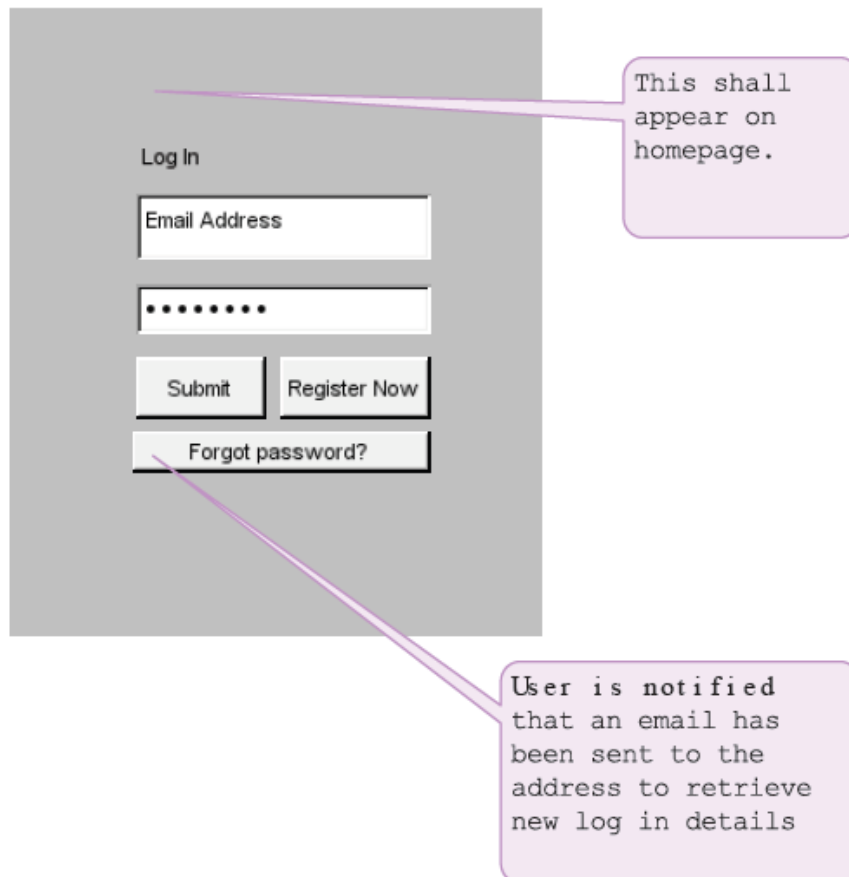
### Alternative Flow

1. Trigger: User selects “Forgot Password?” and link to reset password is sent to the email address.

Divergence point: 1 – 4

Convergence point: User resets password. Starts again at point 1.

### Interface screenshot



### System Use Case: View Profile

**User story:** As a registered user I want to be able to view my profile.

### Basic Flow

1. A registered user views personalised profile.
2. The user may submit a paper through the “Author” tab, and thus become an author
3. The user may not access the “Reviewer” or “Committee Member” tabs until those roles are assigned by the chairperson.

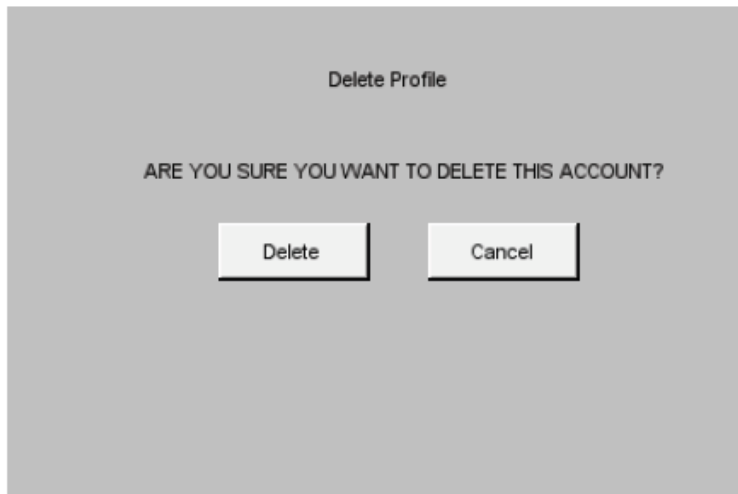
### **System Use Case: Delete Profile**

**User story:** As a registered user I want to be able to delete my profile.

#### **Basic Flow**

1. User selects delete profile
2. User confirms DELETE or may CANCEL.

#### **Interface screenshot**



### **System Use Case: Submit Paper**

**User story:** As an Author, I want to be able to submit a paper, so that I can partake in the conference.

#### **Basic Flow**

1. Author selects Paper Submission
2. Author browses for word file or document to upload
3. Author Selects Upload
4. Author submits selection

## Interface screenshot

Paper Submission

Title:

Metadata; Keywords:

List of Current [Papers](#)

### System Use Case: View status of Paper

**User story:** As an Author, I want to view my status, so that I can see when my paper has been reviewed and if it is part of the conference.

#### Basic Flow

1. An author accesses their profile
2. The author checks the status of the paper next to its title.

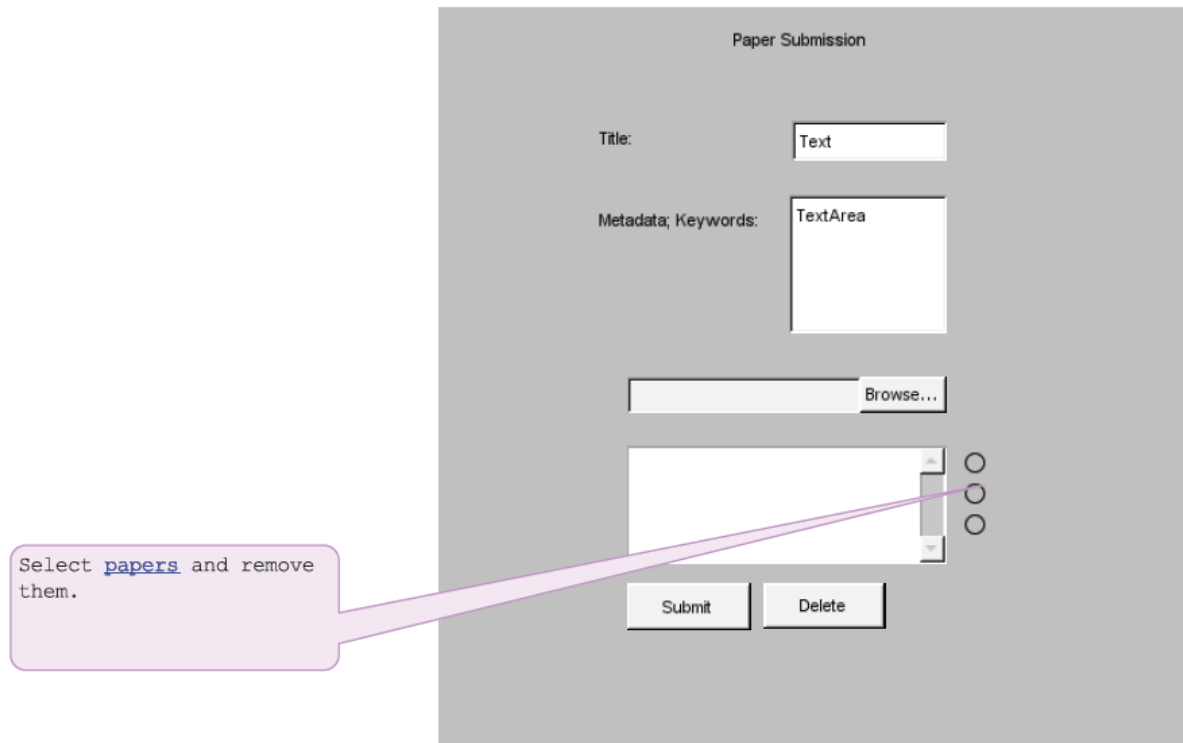
### System Use Case: Update Paper

**User story:** As an Author, I want to be able to update or delete my paper, so that I can make the appropriate changes.

#### Basic Flow

1. Author goes to submit paper.
2. Author selects the make changes option.
3. Author browses for new file or document.
4. Author selects the paper he would like to overwrite (System saves original).
5. Author makes changes and exits.

## Interface screenshot



### System Use Case: View Assigned Papers

**User story:** As a Reviewer, I want to view my assigned papers, so that I can review the papers.

#### Basic Flow

1. Reviewer signs in.
2. Reviewer clicks on "Review" tab.
3. Reviewer browses list of assigned papers.

### **System Use Case: Submit Review**

**User story:** As a Reviewer, I want to submit reviews, so that I can share my view on the paper.

#### **Basic Flow**

1. Reviewer selects paper to review from list.
2. Reviewer reads through article.
3. Reviewer selects a rating.
4. Reviewer may submit or cancel the review.

#### **Alternative Flow**

Trigger: Reviewer decides to write a comment.

Divergence Point: 3.

Convergence Point 4.

#### **Interface screenshot**

The screenshot shows a web interface for submitting a review. At the top, it says "Author's Name and title". Below this is a large rectangular box labeled "Article displayed here." which is currently empty. Underneath the article box is a "Rating" section with a dropdown menu. Below the rating is a text input field labeled "Comments". At the bottom of the form are two buttons: "Submit" and "Cancel".

### **System Use Case: Update Review**

**User story:** As a Reviewer, I want to update my reviews, so that I can make the appropriate changes.

#### **Basic Flow**

1. Reviewer signs in.
2. Reviewer views “Review” tab
3. Reviewer clicks the “Edit Review” button
4. The system checks if it is before the deadline.
5. If it is, reviewer re-uploads the updated review.

#### **Alternative Flow**

**Trigger:** The deadline for submitting reviews is past.

Divergent point: 4

1. The system displays a message that the deadline has already been reached for reviews.

### **System Use Case: Assign Papers**

**User story:** As a Member of Committee, I want to assign papers, so that I can ensure there are no conflicts of interest.

#### **Basic Flow**

1. Committee member goes to the assign paper option
2. Committee member scrolls through the list of papers.
3. Committee member selects a paper.
4. Committee member adds/removes a reviewer to the paper.

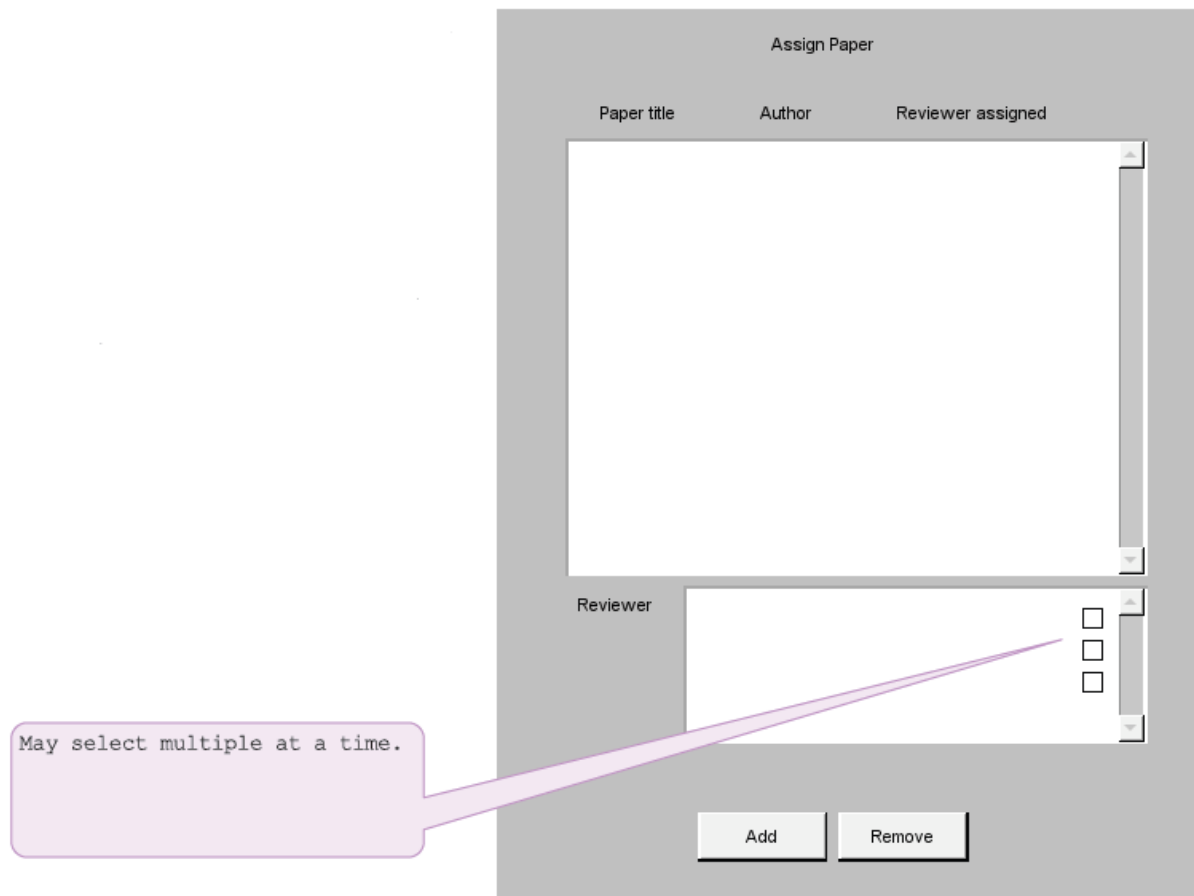
#### **Alternative Flow**

Trigger: Committee member views a reviewer's profile.

Divergence point: 4

Convergence point: 4

## Interface screenshot



### System Use Case: View ALL Papers

**User story:** As a Member of Committee, I want to view papers, so that I can see the papers adhere to conference format.

#### Basic Flow

1. Committee member signs in.
2. Committee member clicks on “Committee Member” tab.
3. Committee member views list of all submitted papers.

### System Use Case: View Reviews

**User story:** As a Member of Committee, I want to view reviews, so that I can put forward my score.

#### Basic Flow

1. Committee member signs in.
2. Committee member clicks on “Committee Member” tab.
3. Committee member views list of all submitted reviews.



### **System Use Case: Update Conference Website**

**User story:** As Chairperson, I want to manage or update conference website, so that I change appropriate conference details.

#### **Basic Flow**

1. Chairperson makes changes to the conference information.
2. Chairperson saves changes and exits.

#### **Interface screenshot**

The screenshot displays a web interface for updating conference details. The main area has a light gray background. At the top right, there is a button labeled "Edit this page?". Below this, the text "ISISM Conference 2016" is displayed. Underneath, the dates "2016 - 01 - 15" are shown. A section labeled "Headline" contains a large white rectangular text input field with the placeholder text "Information about conference.". Below the main interface, a separate gray dialog box is shown. It contains the question "Are you sure you want to save changes?" and two buttons: "Save" and "Cancel".

### **System Use Case: View Registered Members**

**User story:** As Chairperson, I want to view registered members, so that I can see everyone for administration purposes or check for conflicts.

#### **Basic Flow**

1. Chairperson signs in.
2. Chairperson clicks on “View all users” tab

### **System Use Case: Assign Roles to Registered Members**

**User story:** As Chairperson, I want to manage or update registered members, so that I can make appropriate changes.

As a Chairperson, I want to assign reviewers, so that I can papers will be reviewed for the conference.

As a Chairperson, I want to assign programme committee members, so that I can select the committee for papers to be accepted.

#### **Basic Flow**

1. Chairperson goes to user list.
2. Chairperson selects user that he wishes to edit.
3. Chairperson selects Change Privileges
4. Chairperson selects 'Assign Committee Member' or 'Assign Reviewer'
5. Chairperson Saves Changes

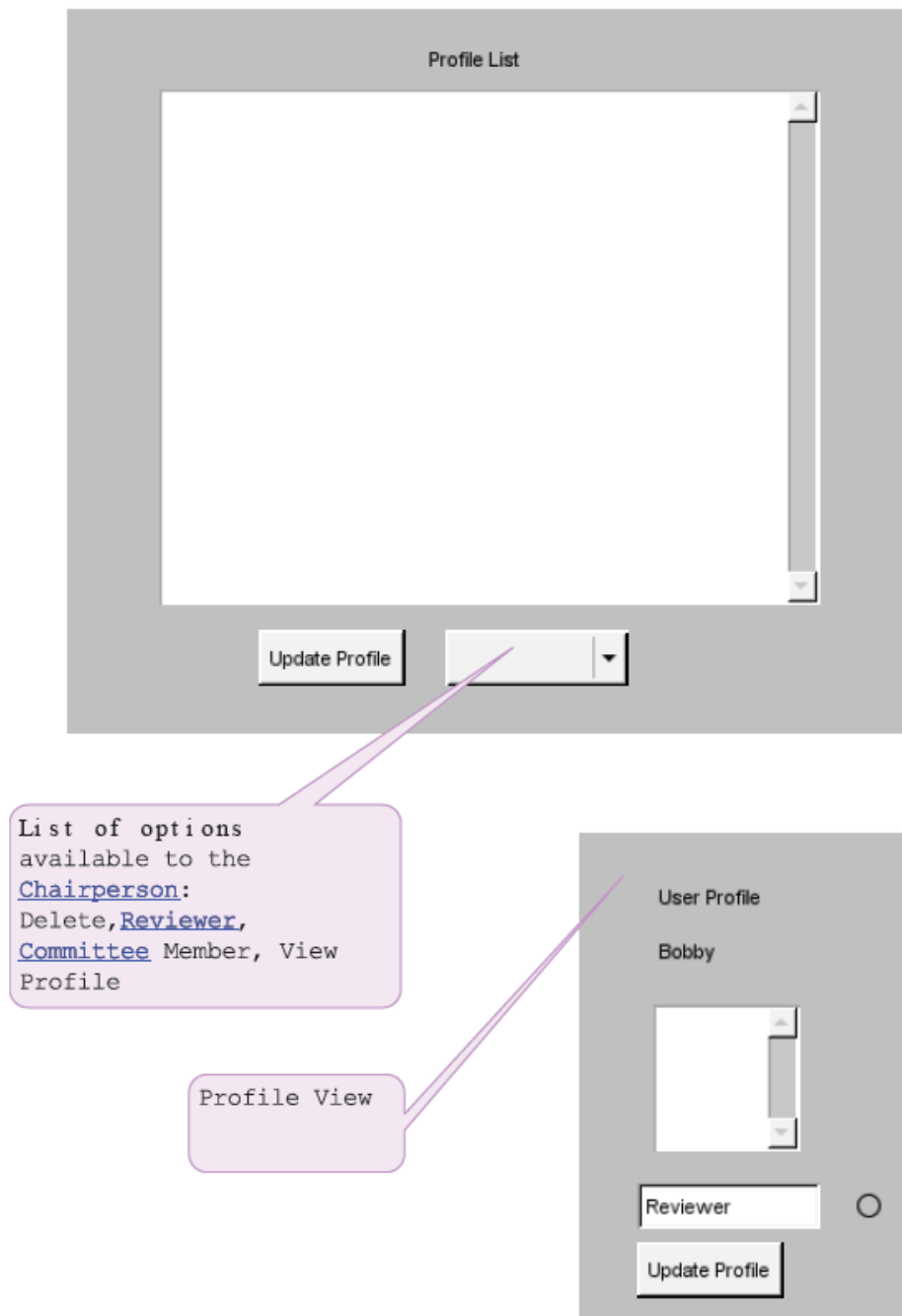
#### **Alternative Flow**

Trigger: Chairperson double clicks on User. Is taken to their profile.

Divergence Point: 2

Convergence Point: 5.

## Interface screenshot



### **System Use Case: Accept/Reject Papers**

**User story:** As a Chairperson, I want to accept or reject papers, so that I can make final decisions on the conference.

#### **Basic Flow**

1. Chairperson selects reviewed papers.
2. Chairperson clicks on paper.
3. Marks the paper with accepted or rejected.
4. System prompts whether he wants to save changes.

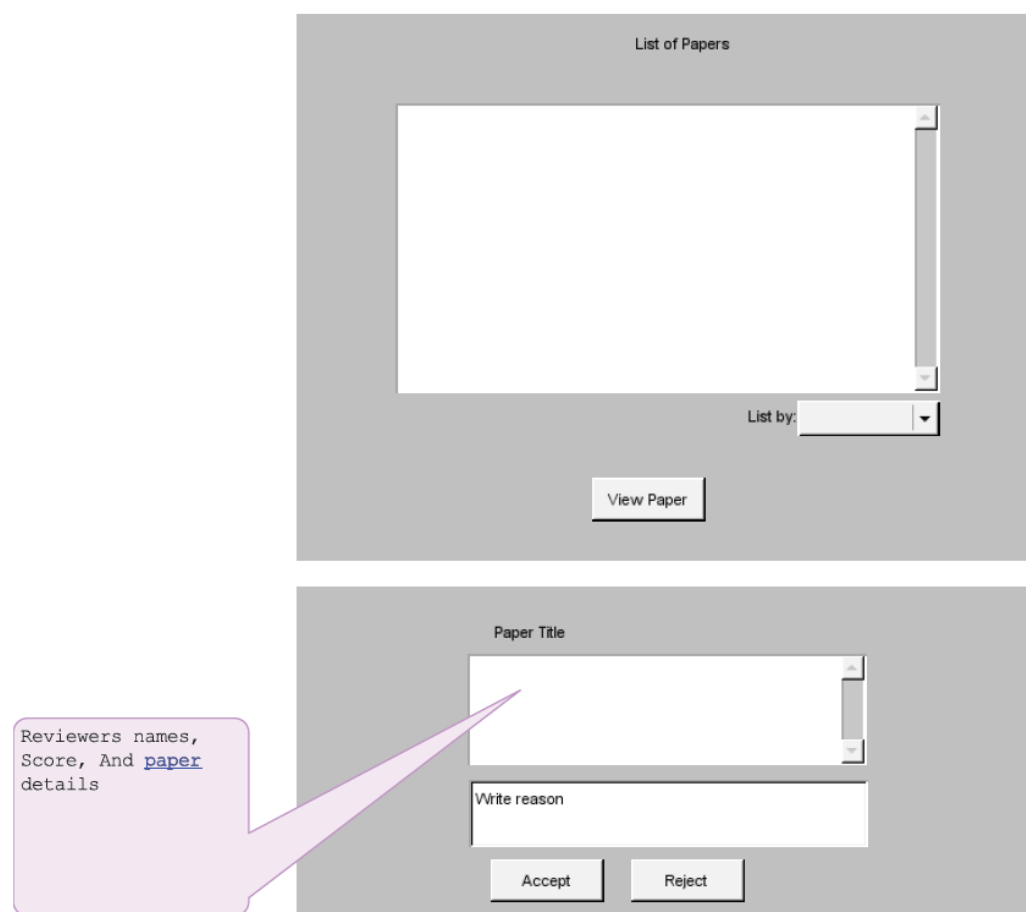
#### **Alternative Flow**

Trigger: Before clicking on a paper, Chairperson can select 'List by'. This will save time and ensure he accepts according to average score.

Divergence Point: 2.

Convergence Point: 2

#### **Interface screenshot**

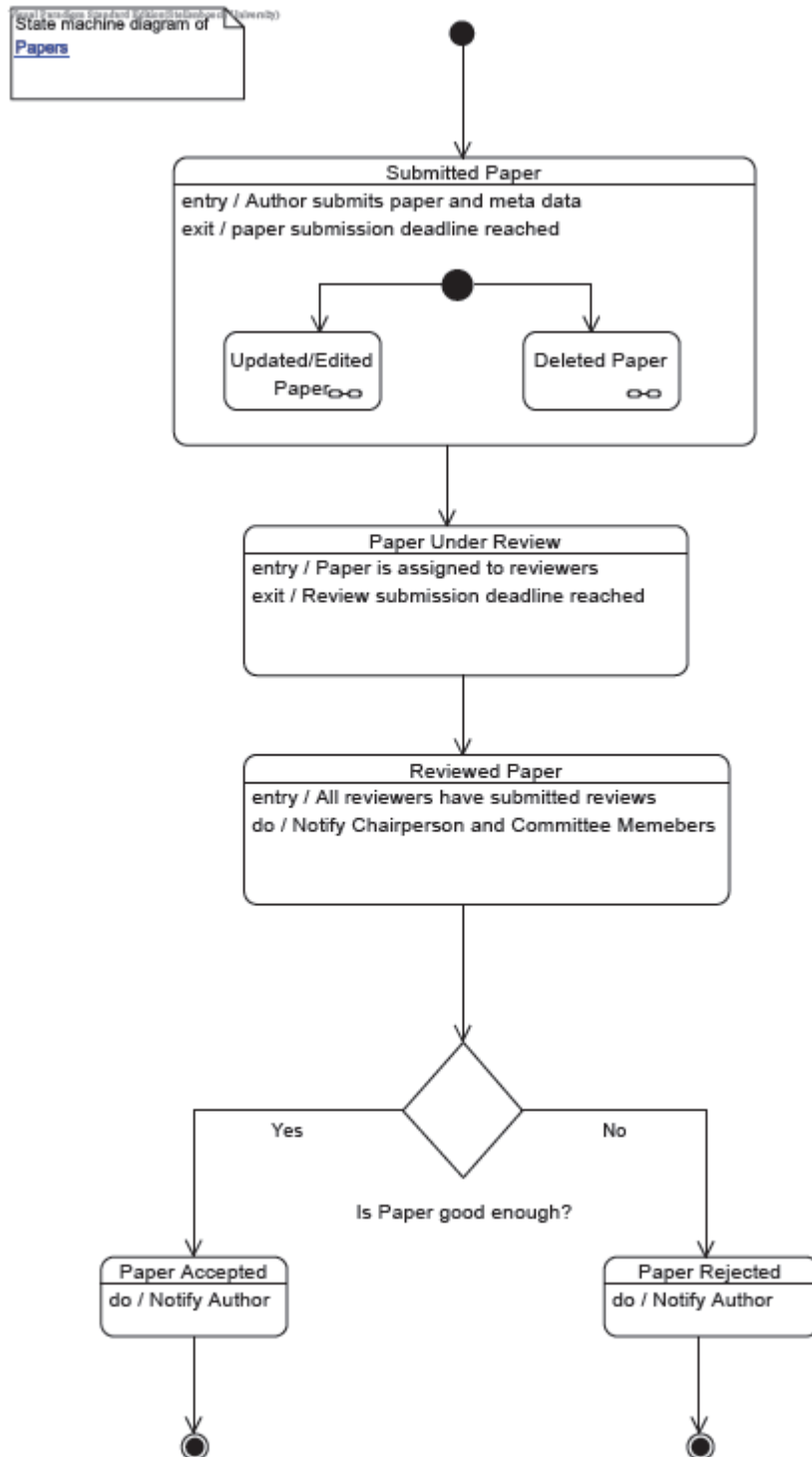


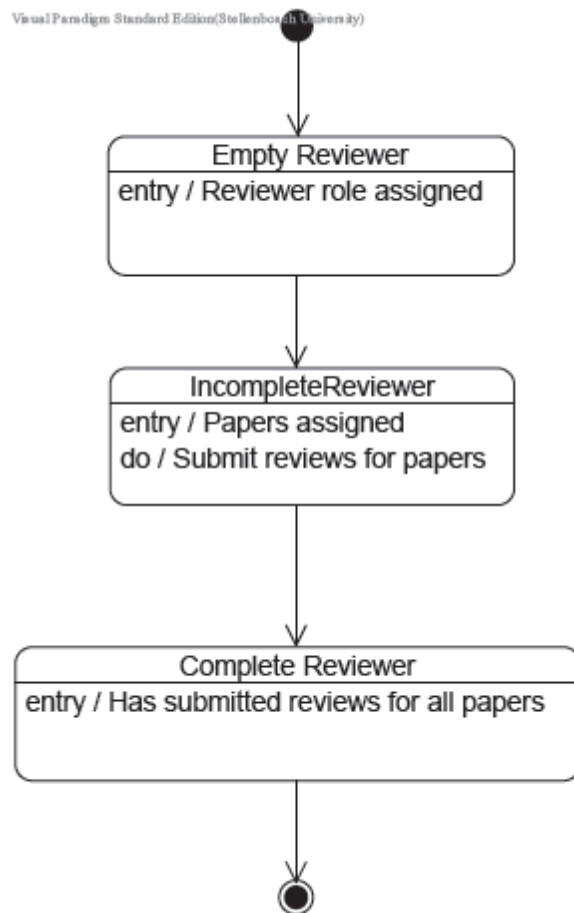
**System Use Case: Manage Conference**

**User story:** As a Chairperson, I want to set deadlines, so that I can make sure all conference related tasks and the conference has a timeline and structure.

## State-Machine Diagrams

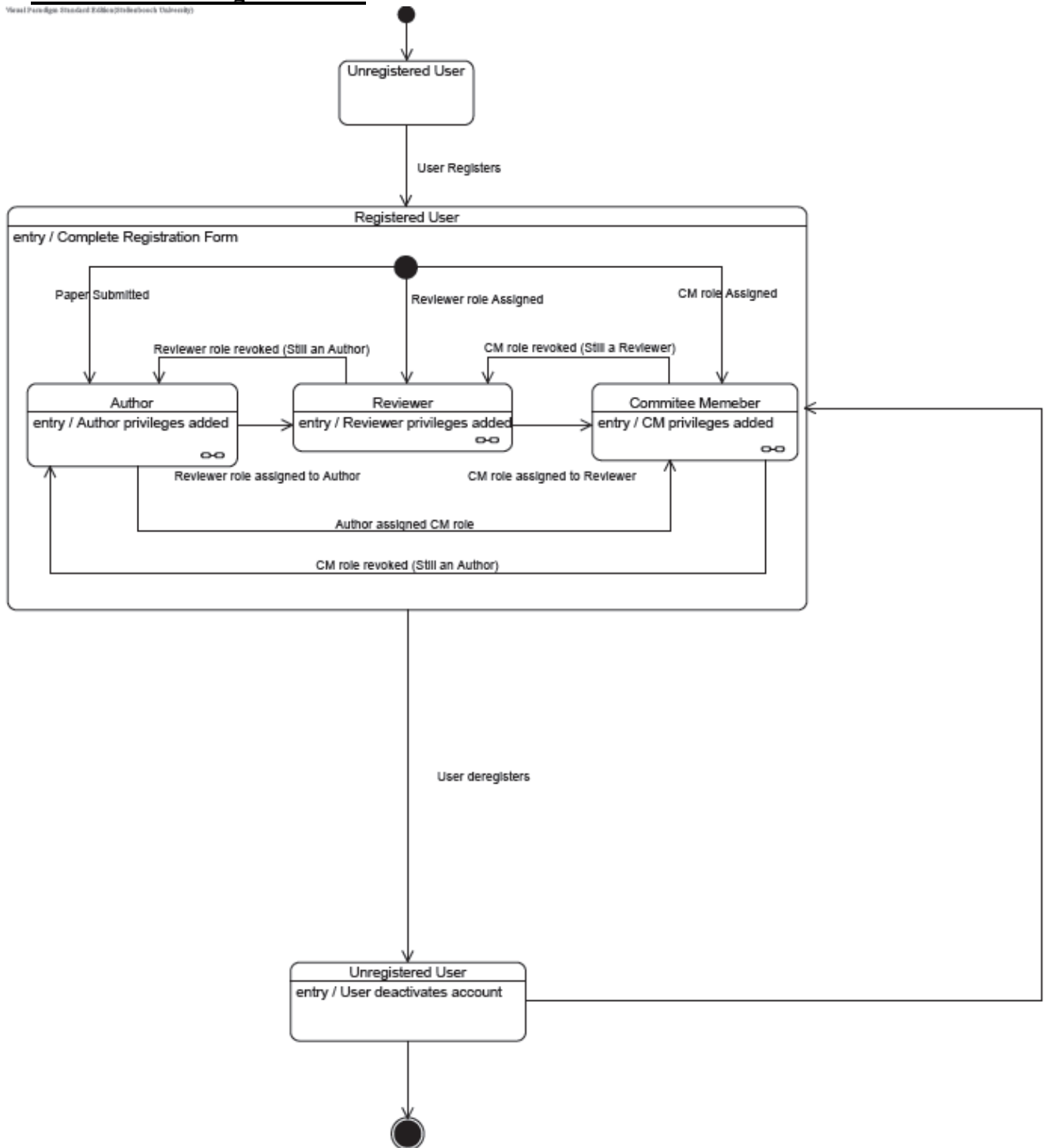
### State-Machine Diagram: Papers



**State-Machine Diagram: Reviewers**

**State-Machine Diagram: Users**

View Full Page: Printed Edition (for Researcher Only)





## **Non-Functional Requirements**

### **Performance Requirements**

- **Stress Requirements:** The system must be able to handle 20 concurrent users, submitting/downloading papers, and 10 concurrent users submitting and accessing reviews.
- **Response-Time Requirements:** Response time should be under 2 seconds
- **Throughput Requirements:** TBD

### **Usability Requirements**

The system must be user friendly and contain helpful tips on how the system works. a Help document will accompany the software.

### **Security Requirements**

- Only the chairperson should be able to: assign roles and mark paper as accepted/rejected.
- Only Committee members should be able to view all papers and reviews, and assign papers to reviewers.
- Only reviewers should be able to submit reviews and see the papers assigned to them.
- Authors should only be able to see the papers they have submitted.

### **Volume and Storage Requirements**

The system needs to support a total paper load of 200 papers per conference.

The system needs to support a total load of 300 profiles. These include the profiles of authors, reviewers and committee members.

### **Configuration Requirements**

The system will be PC-compatible.

It will be supported by: Linux  
Windows 7, 8 and 10

It will be compatible with the following web browsers: Internet Explorer  
Mozilla Firefox  
Google Chrome

### **Compatibility Requirements**

The software must be compatible with the payment system used.

### **Reliability Requirements**

There will be no system down time.

Maintenance will be done during non-working hours and must not exceed one hour.

**Backup/Recovery Requirements**

Daily backups of the database to the hard drive will be taking place.  
There will be weekly backups of the entire system.

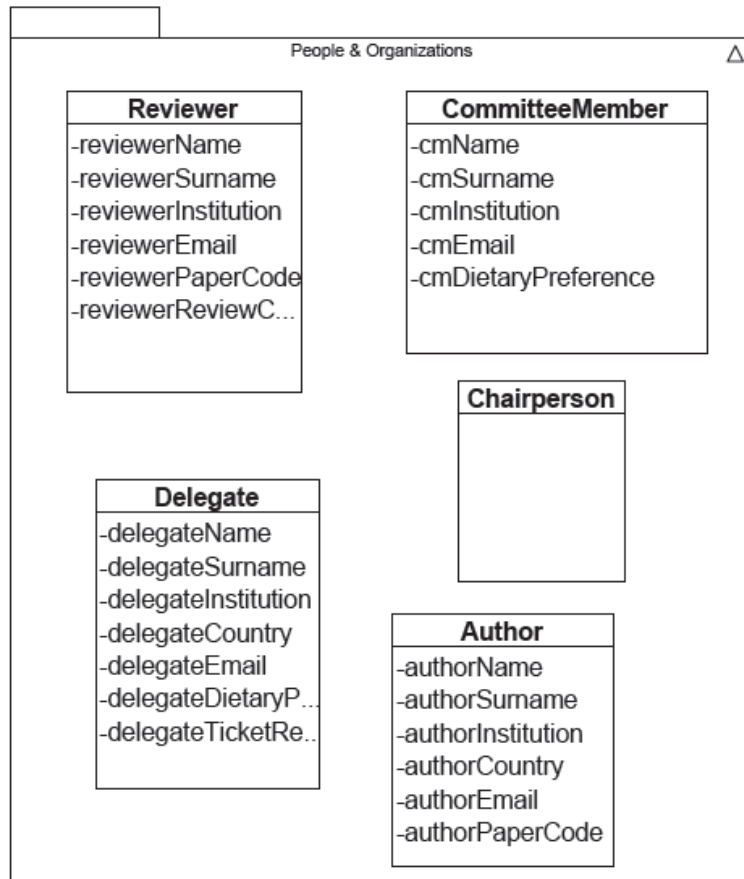
**Training Requirements**

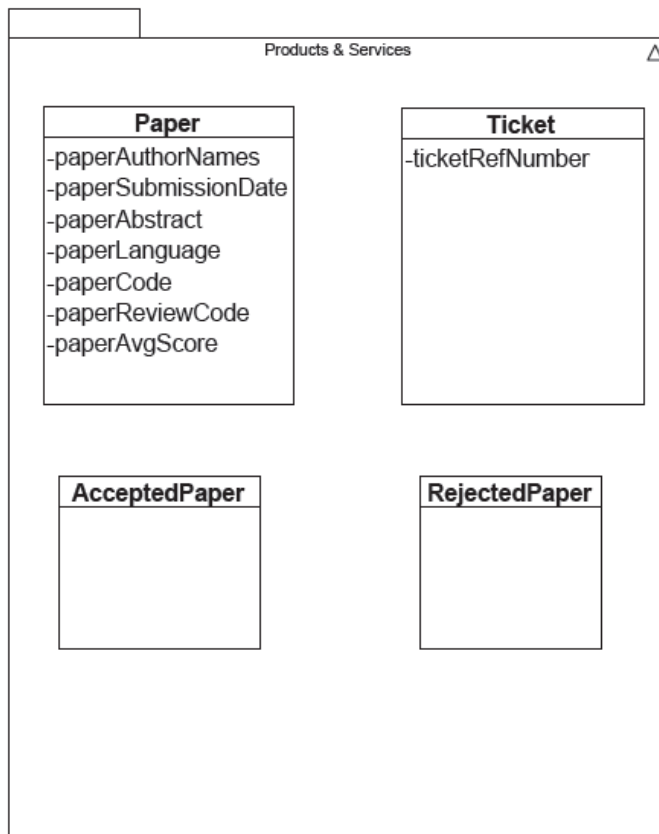
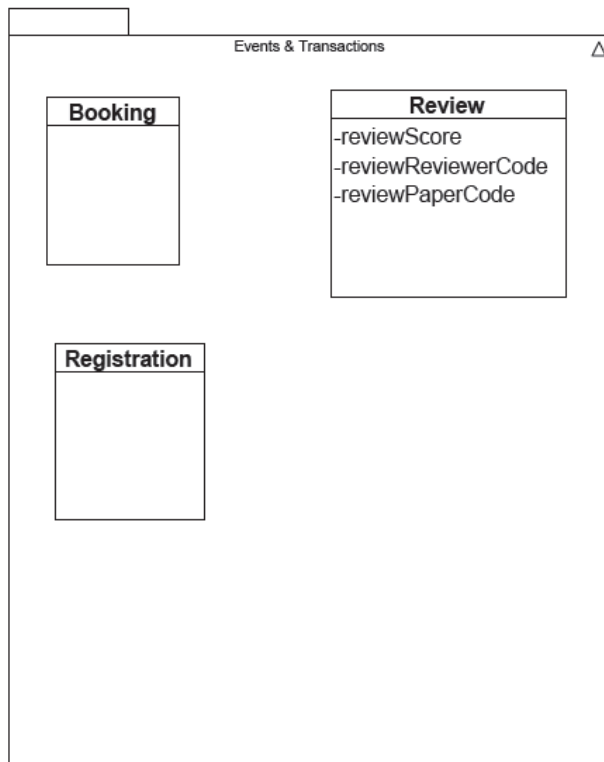
A help document containing tips on how to use the system will be provided to the user.

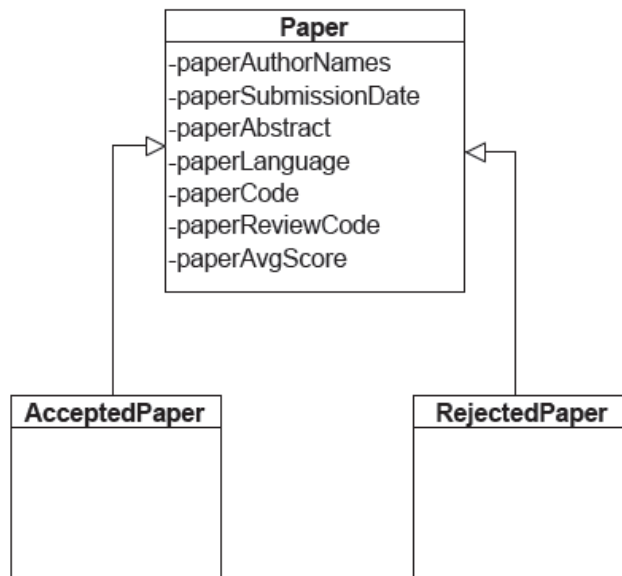
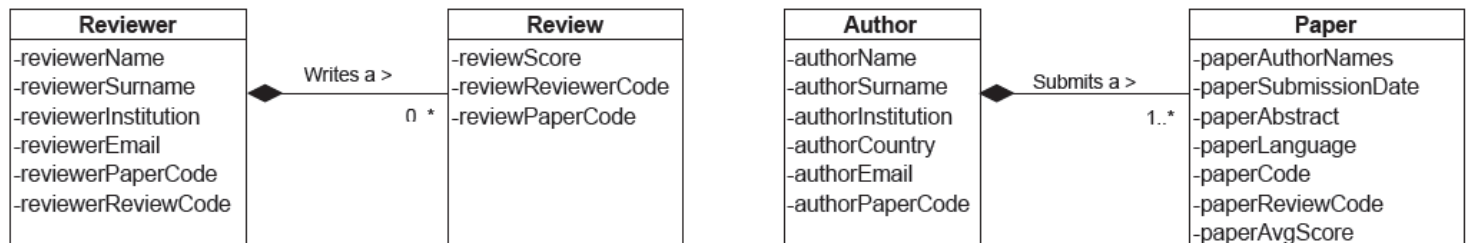
## Structural Model

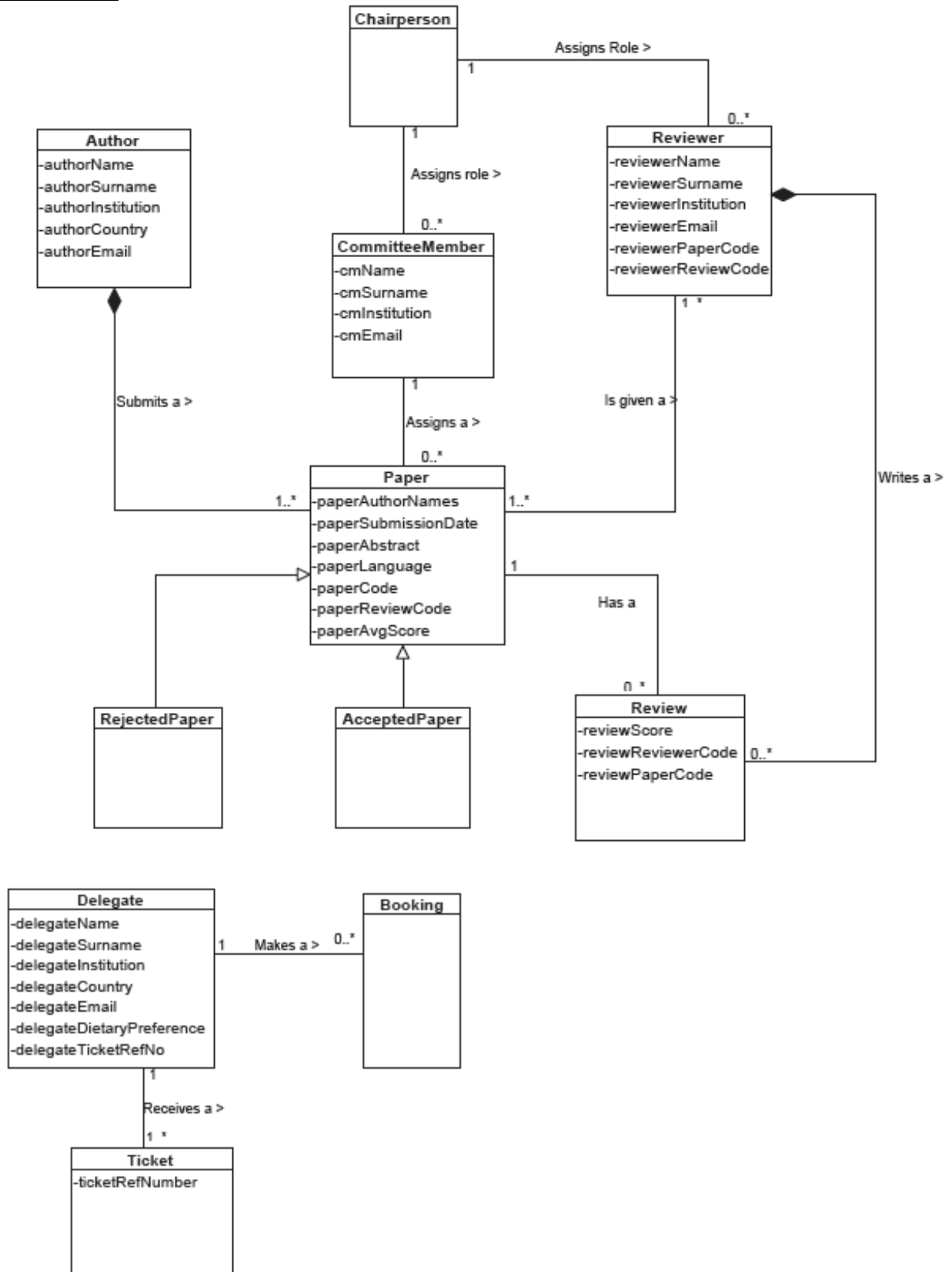
### Class Diagrams: Entity Classes

#### Package: People and Organizations



**Package: Products and Services****Package: Events/Transactions**

**Specialization/Generalization****Whole/Part Relationships**

Associations

## **Test Plan**

1. Develop test scenarios for the requirements-based testing. Conduct white-box testing to check whether the fields and forms function as intended.
2. Conduct the requirements-based testing. Prove whether the requirements comply. Determine techniques to be used in black-box testing, such as structured testing guidelines.
3. Perform system testing. Using regression, integration, volume and stress tests.
4. Conduct user-acceptance tests. The end-users interact with the system and see whether it is user-friendly and meet their specifications.

## **Implementation Plan**

### **Training**

- The developers of the system is responsible for providing training.
- **Training audience:** Chairperson, Committee Members, Reviewers
- **Forum:** Three individual one hour sessions for the chairperson and two groups.

### **Rollout**

Notify Chairperson when system is ready for use.

### **Sign-Off**

## **Glossary**

<b>Name</b>	<b>Aliases</b>	<b>Description</b>
Author	Authors	Is an individual who has submitted a paper.
Book conference		Fill out form on website in order to purchase ticket
Chairperson	Admin	In charge of site and creating the programme
Committee		Members who work for the conference
paper	submission, papers	The article or research journal an author submits.



## **Reference List**

Podeswa, H. 2010. UML for the IT Business Analyst, Course Technology, Boston.

Interviews with various shareholders.