SAM2017

Design Document

RELEASE 1

GROUP 5 – YOU KNOW WHO

Table of Contents

[Architectural Overview 2](#_Toc463187605)

[Class Diagram 7](#_Toc463187607)

[Sequence Diagram(s) 8](#_Toc463187608)

[State Machine 12](#_Toc463187609)

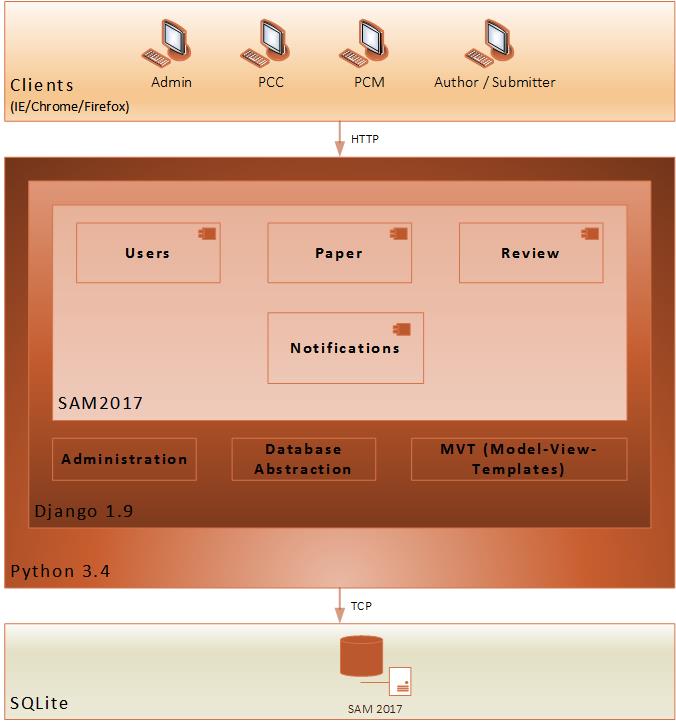
[Design Rationale](#_Toc463187606) 16

|  |  |  |  |
| --- | --- | --- | --- |
| ***Revision Number*** | ***Revision Date*** | ***Summary of Changes*** | ***Author(s)*** |
| 1 | 12 Oct 2016 | Initial Content | Adriana Sejfia  Nasir Safdari  Umang Garg  Smruthi Gadenkanahalli |

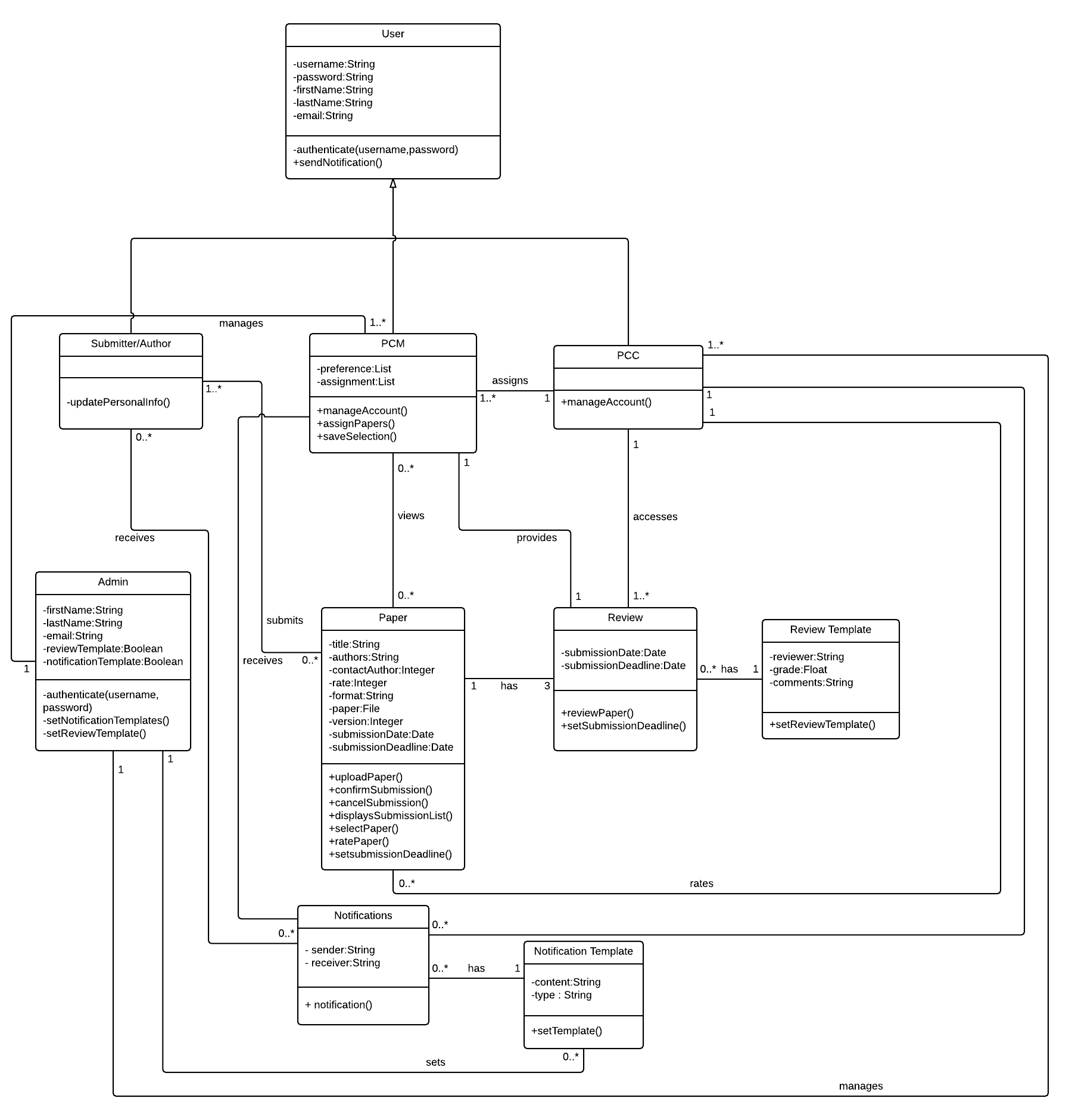
# Architecture Diagram

This diagram represents the system and components involved in the application and their relationship. The components and subsystems are placed in different layers namely, Presentation layer, Business/Domain Layer and Data layer.

# arch1.jpg



# Class Diagram



*Figure 1 Class Diagram for SAM2017*

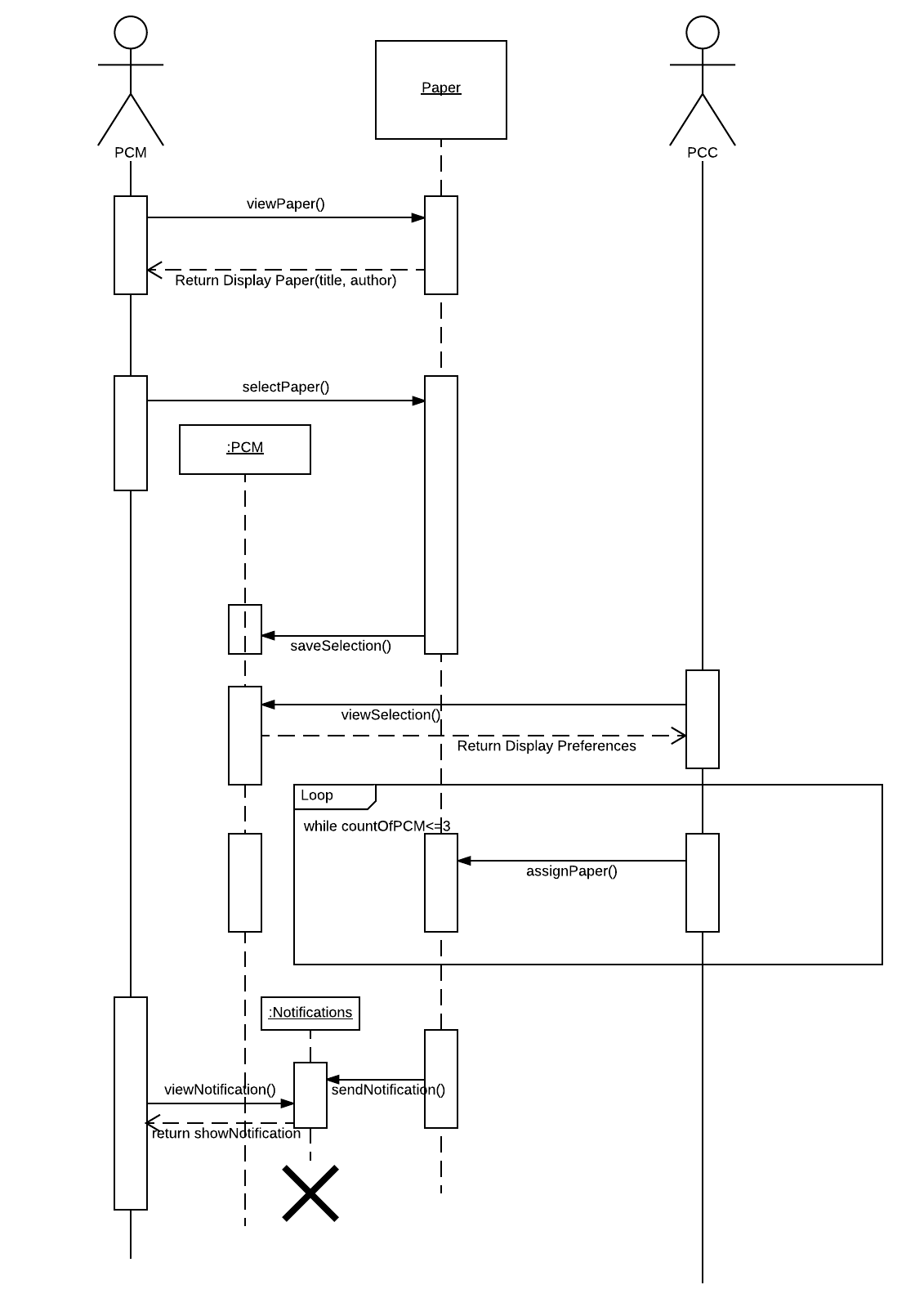
# 

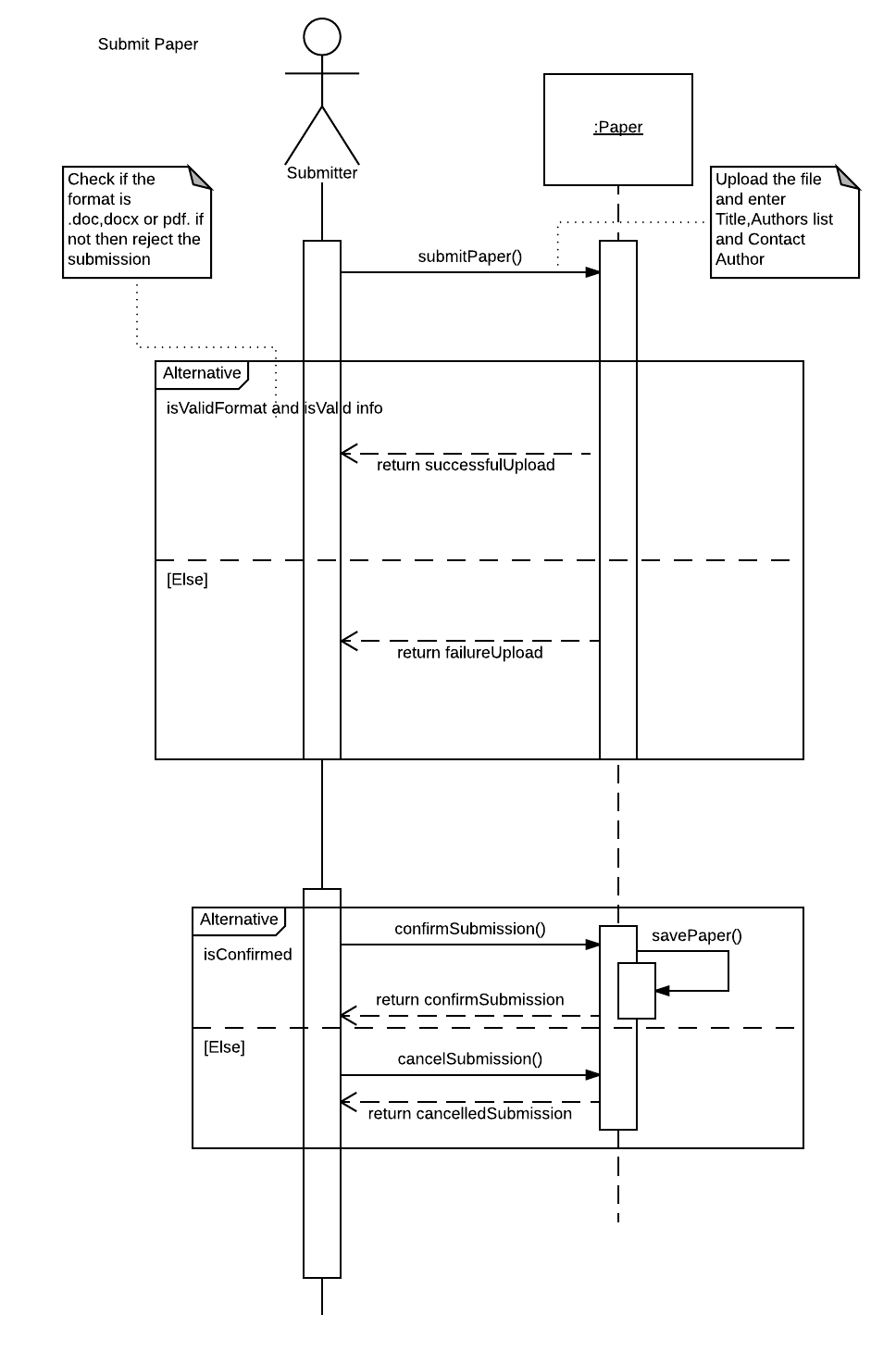
# Sequence Diagrams

1. Registration and Login

# Register And Login - Page 1 (1).png

1. View, Select, Assign Paper



C. Submit Paper

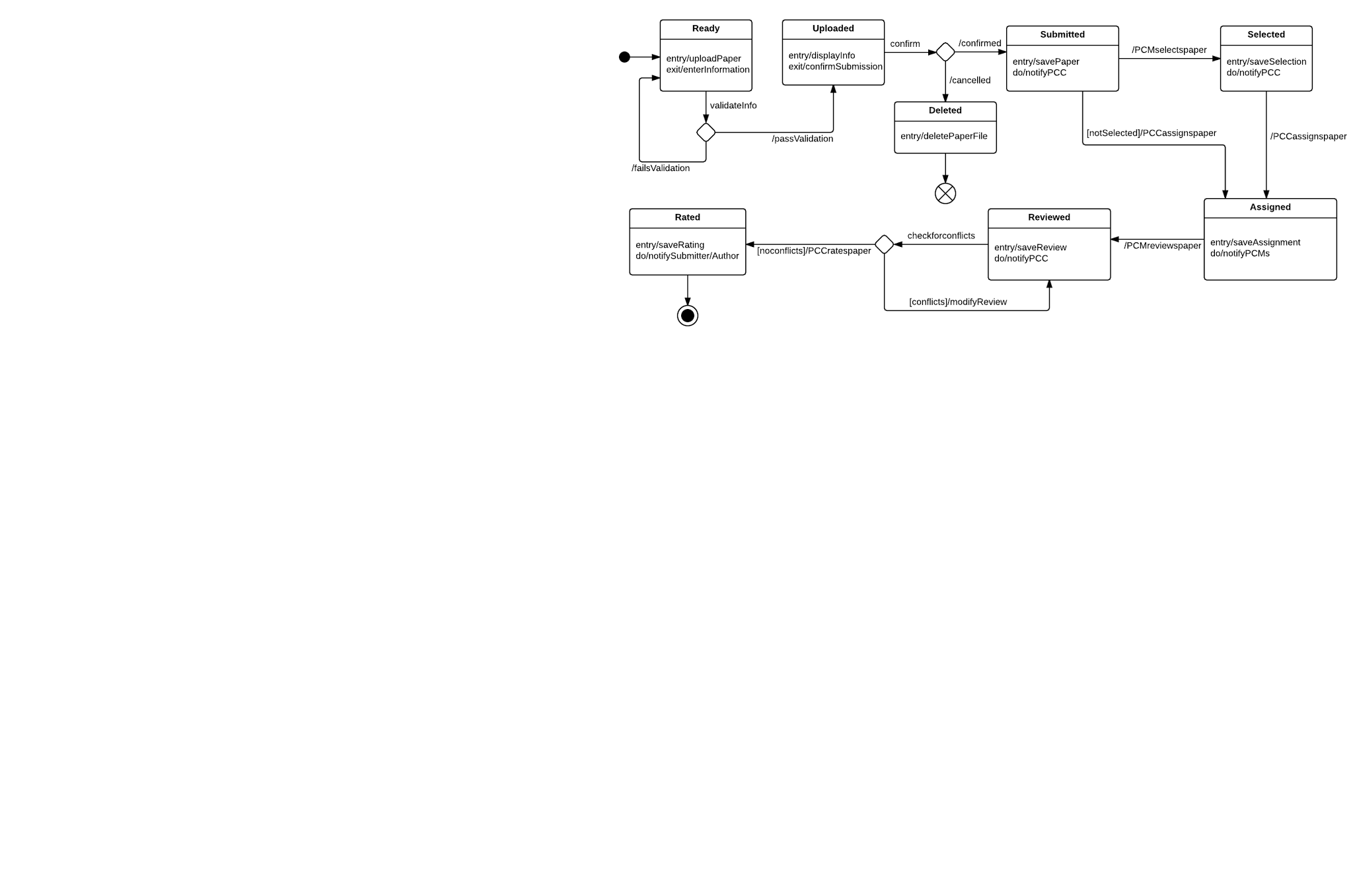
# 

# State Machine Diagram

A. Registration

# 

B. Upload, Review and Rate



# Design Rationale

Technology Rationale

As initiated in the requirement gathering phase, Django was selected as the web framework to implement this web application. Django provides the following features:

* A Python based web-framework for rapid development
* Template based front-end development.
* Built-in ORM support
* Automatic admin UI generation

SAM 2017 is built on a classic layered architecture, where each layer can only communicate with the layer that it is in-between. This not only reduces coupling, but also acts as a security mechanism.

Class Diagram Rationale

During our regular discussions in our meetings, it was identified that if we divide the Paper class from the Review class, we might be able to solve future problems for making changes and invoking other methods by other actors on these two classes. In addition, we had interesting ideas for having some more classes, however, while planning for implementation we decided to reduce the number of

The ‘User’ class generalize three other classes – Author, PCC, and PCM. Using generalization, we abled our design to simplify the process of distributing attributes among those submissive classes.

Alternatives

Instead of Django, Microsoft or PHP based technologies could have been utilized to build the solution. Both Microsoft and PHP, have web-frameworks, like ASP.net and Code Igniter that can be utilized to build the solution. Further, instead of SQLite, MySQL and MS SQL can be utilized for data storage.