Software Modeling

SAM2017 Project

Release 2

Group 6

Student Name: Ian Shoenberger

Venkat Nitin Panaganti

Wajdi Aljedaani

Muhammad Fazalul Rahman

Instructor Name: Dr. J.Scott Hawker

Contents

Use case diagram

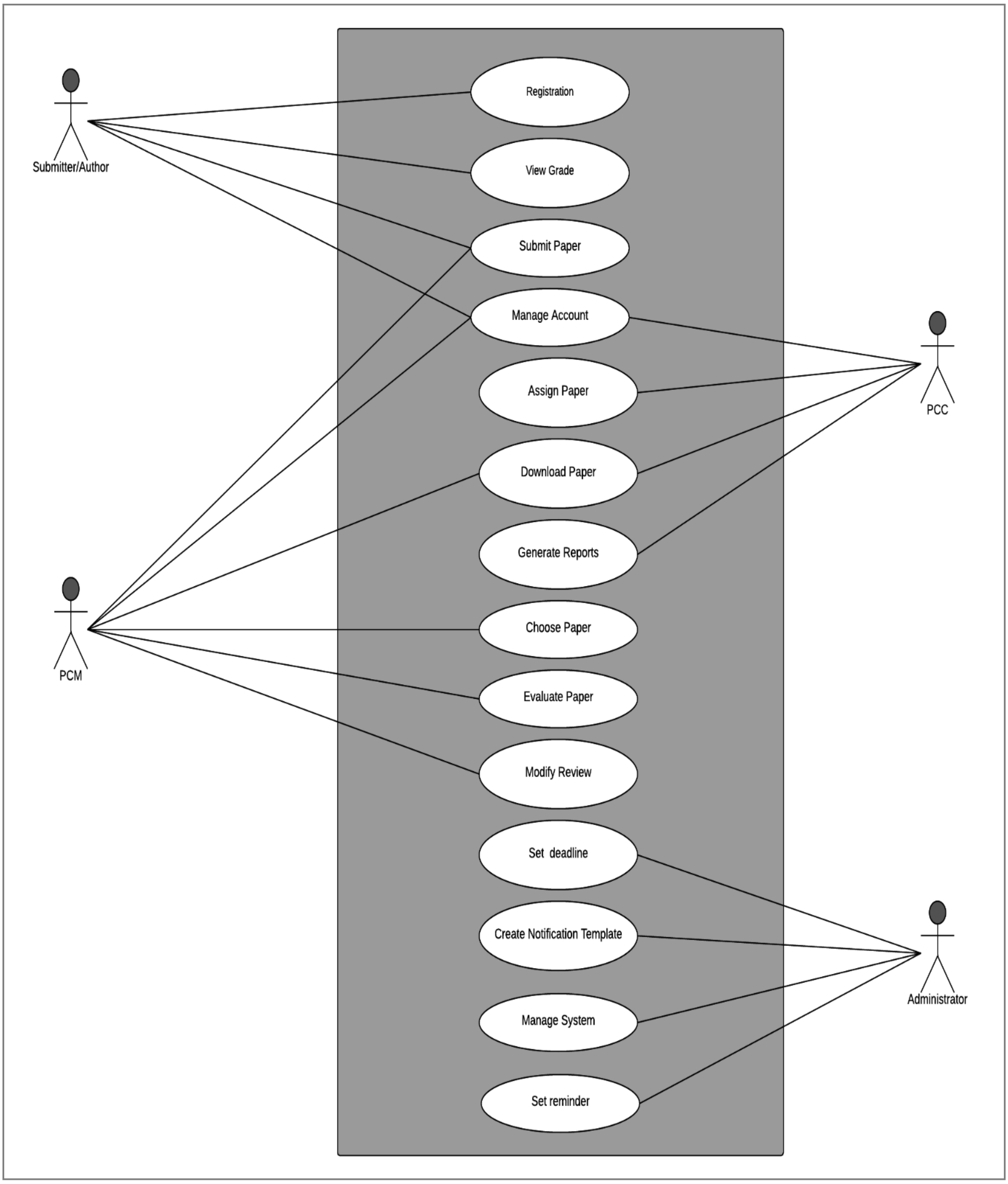

Figure 1 Use case diagram

Use case list

Use Case ID	Primary Actor	Use Cases Name
UC01	Submitter/Author	Registration
UC02	Submitter/Author	View Grade
UC03	Submitter/Author PCM	Submit paper
UC04	PCC	Assign paper
UC05	PCM PCC	Download paper
UC06	PCC	Generate report
UC07	PCM	Choose paper
UC08	PCM	Evaluate paper
UC09	PCM PCC	Modify review
UC10	Administrator	Set deadline
UC11	Administrator	Create notification template
UC12	Administrator	Manage system
UC13	Submitter/Author PCM PCC	Manage account
UC14	Administrator	Set reminder

Use Cases Description

Use Case ID:	UC01		
Use Case Name:	Registration		
Created By:	Venkat Nitin	**Last Updated By:**	Team 6
Date Created:	09-24-2015	**Date Last Updated:**	09-24-2015
Overview:	Registrant shall be able to provide personal information to the SAM2017 upon registering.		
Actors:	Submitter/Author		
Type:	Primary		
Preconditions:	- SAM2017 has been setup and configured. - SAM2017 is running and open for registrations. - Registrant has accessed website via URL		
Main Flow:	1. Registrant clicks on the Register from home page. 2. SAM2017 shows the registration page to registrant and ask for the personal information (first name, last name, username, email and password). 3. Registrant fills out his/her personal information and click on submit. 4. SAM2017 verifies required information is provided and saved it in the SAM2017. 5. SAM2017 confirm the registration and redirect the registrant to home page.		
Alternative Flows (1):	After Step (2) registrant clicks on Cancel button. 1. SAM2017 redirects registrant to main page.		
Alternative Flows (2):	After Step (4) if the registrant has entered an invalid information such as email, SAM2017 shows a message that personal information are invalid. 1. SAM2017 redirect registrant to step 2.		
Post conditions:	Registrant has registered successfully in the SAM2017.		
Post conditions alternative Flows (1):	SAM2017 has cancelled the registration successfully which made by registrant.		
Post conditions alternative Flows (2):	SAM2017 has showed the message to registrant successfully.		
Priority:	1		

Use Case ID:	UC02		
Use Case Name:	View Grade		
Created By:	Wajdi Aljedaani	**Last Updated By:**	Team 6
Date Created:	11-01-2015	**Date Last Updated:**	12-06-2015
Overview:	The Author shall be able to view the grade submitted by the PCC.		
Actors:	Submitter/Author		
Type:	Primary		
Preconditions:	- The author has submitted a paper. - The PCMs have submitted their ratings. - The PCC has compared the ratings and submitted the grade and review for the paper. - The author must be logged in to the SAM2017.		
Main Flow:	1. The author chooses to view grade for a particular paper. 2. SAM2017 displays the grade and comments that were submitted by the PCC and PCMs. 3. The author views the comments and final grades.		
Alternative Flows (1):	After Step (1) SAM2017 displays a message showing that the grades aren't available yet.		
Post conditions:	The user has successfully viewed the grade for his paper.		
Post conditions alternative Flows (1):	The user could not view his / her grade.		
Priority:	1		

Use Case ID:	UC03		
Use Case Name:	Submit paper		
Created By:	Venkat Nitin	**Last Updated By:**	Team 6
Date Created:	09-28-2015	**Date Last Updated:**	12-06-2015
Overview:	Submitter/Author or PCM shall be able to submit paper to the SAM2017.		
Actors:	Submitter/Author, PCM		
Type:	Primary		
Preconditions:	- Author must be logged in to the SAM2017.		
Main Flow:	1. Author or PCM clicks on the submit paper option. 2. SAM2017 displays submit paper page. 3. Author or PCM fill out the template information of submission paper (title, authors and format of paper) with proper format (PDF or Microsoft Word). 4. SAM2017 verifies the file type. 5. SAM2017 reloads the same page with the updated list of submissions.		
Alternative Flows (1):	After Step (4), if the Author or PCM upload an invalid format, SAM2017 displays message that the format is invalid. 1. SAM2017 redirects Author to Step 2.		
Alternative Flows (2):	After Step (3) Author or PCM clicks on Cancel button. 1. SAM2017 redirects Author to main page.		
Post conditions:	Author or PCM has submitted his paper successfully to the SAM2017.		
Post conditions Alternative Flows (1):	SAM2017 has showed the message to the Submitter/Author or PCM successfully.		
Post conditions Alternative Flows (2):	SAM2017 has cancelled to submit paper successfully that made by the Submitter/Author or PCM.		
Priority:	1		

Use Case ID:	UC04		
Use Case Name:	Assign paper		
Created By:	Ian Shoenberger	**Last Updated By:**	Team 6
Date Created:	09-24-2015	**Date Last Updated:**	12-06-2015
Overview:	PCC shall be able to assign paper to a three PCM's to review.		
Actors:	PCC, PCM		
Type:	Primary		
Preconditions:	- PCC must be logged in to the SAM2017. - Submission paper must be in the SAM2017.		
Main Flow:	1. The PCC navigates to the home page. 2. SAM2017 displays the list of papers. 3. PCC selects a paper that has not yet been assigned to three PCM's. 4. SAM2017 display details about the paper and shows a list of available PCM's as well as an indication of which PCM's preselected the paper. 5. PCC selects three PCM's from the list. 6. PCC submits their assignments.		
Alternative Flows (1):	After Step (2) if there are not enough PCM's (less than three), SAM2017 will displays message that there are not enough PCM's in the SAM2017.		
Alternative Flows (2):	After Step (3) PCC chooses to cancel. 1. SAM2017 redirects submitter to main page.		
Post conditions:	SAM2017 assigned the paper successfully and records the assignment. SAM2017 gives privilege to the selected PCM's to view the paper in its entirety.		
Post conditions alternative Flows (1):	SAM2017 has showed the message to the PCC successfully.		
Post conditions alternative Flows (2):	The SAM2017 has cancelled the assignment process.		
Priority:	1		

Use Case ID:	UC05		
Use Case Name:	Download paper		
Created By:	Wajdi Aljedaani	**Last Updated By:**	Team 6
Date Created:	09-24-2015	**Date Last Updated:**	12-05-2015
Overview:	PCC or PCM shall be able to download the submitted paper in the SAM2017.		
Actors:	PCC, PCM		
Type:	Primary		
Preconditions:	- PCC or PCM must be logged in to the SAM2017. - Submitted paper must be in the SAM2017.		
Main Flow:	1. PCC navigates to the home page. 2. SAM2017 displays all the paper that has been submitted. 3. SAM2017 displays all the list of paper that has submitted. 4. PCC selects the paper that he/she wants to download. 5. SAM2017 downloads the selected paper.		
Alternative Flows (1):	1. F.1 If the user is PCM, SAM2017 displays all the paper that he/she has been assigned to review. PCM selects the paper that he/she would like to download. 1. SAM2017 displays download the selected paper to PCM.		
Post conditions:	PCC has successfully downloaded the paper from SAM2017.		
Post conditions alternative Flows (1)	PCM has successfully downloaded the paper from SAM2017.		
Priority:	2		

Use Case ID:	UC06		
Use Case Name:	Generate report		
Created By:	Muhammad Fazalul Rahman	**Last Updated By:**	Team 6
Date Created:	09-24-2015	**Date Last Updated:**	12-05-2015
Overview:	PCC shall be able to generate reports based on the reviews submitted by the PCM.		
Actors:	PCC		
Type:	Primary		
Preconditions:	- PCC must be logged in to the SAM2017. - PCM must submit reviews to PCC. - SAM2017 has sent a notification to PCC that all the PCM reviews have been submitted.		
Main Flow:	1. PCC chooses to generate report. 2. SAM2017 generates the grade and compiles all the PCM comments based on the PCM reviews. 3. PCC selects a grade and fills out a comment. 4. PCC submits the report.		
Alternative Flows (1):	Not applicable		
Post conditions:	PCC has generated the report of the submitted paper successfully. SAM2017 notifies the Author of the generated report.		
Priority:	2		

Use Case ID:	UC07		
Use Case Name:	Choose paper		
Created By:	Ian Shoenberger	**Last Updated By:**	Team 6
Date Created:	09-24-2015	**Date Last Updated:**	12-06-2015
Overview:	PCM shall be able to choose paper from an available set to tag papers he/she would like to later review.		
Actors:	PCM		
Type:	Primary		
Preconditions:	- PCM must be logged in to the SAM2017. - Submitted paper must be in the SAM2017.		
Main Flow:	1. PCM chooses to view recently uploaded papers. 2. SAM2017 displays all available papers by title name. 3. PCM selects an available paper. 4. PCM may repeat step 3. For as many papers are available. 5. PCM submits his/her selection.		
Alternative Flows:	Not Applicable		
Post conditions:	PCM has chosen paper from the list successfully in the SAM2017. SAM2017 records the selections made by the PCM.		
Priority:	1		

Use Case ID:	UC08		
Use Case Name:	Evaluate paper		
Created By:	Venkat Nitin	**Last Updated By:**	Team 6
Date Created:	09-24-2015	**Date Last Updated:**	12-06-2015
Overview:	PCM shall be able to review the paper which was assigned by PCC. The PCM reads the papers and sends a review to the PCC.		
Actors:	PCM		
Type:	Primary		
Preconditions:	- PCM must be logged in to the SAM2017. - Submitted paper must be in the SAM2017. - Paper must be assigned to PCM's.		
Main Flow:	1. PCM chooses to evaluate the paper. 2. SAM2017 displays a template for evaluation of the assigned paper to PCM. 3. PCM writes a review and rate the paper for the paper that was assigned. 4. PCM submits the evaluation.		
Alternative Flows (1):	Not applicable		
Post conditions:	PCM has evaluated the paper. SAM2017 redirects PCM to the home page. If all three evaluations have been submitted, the PCC is notified.		
Priority:	1		

Use Case ID:	UC09		
Use Case Name:	Modify review		
Created By:	Muhammad Fazalul Rahman	**Last Updated By:**	Team 6
Date Created:	09-24-2015	**Date Last Updated:**	12-06-2015
Overview:	PCC shall be able to find conflicts and notifies the PCM who has to re-review the submission.		
Actors:	PCM, PCC		
Type:	Primary		
Preconditions:	- PCC shall be logged in to the SAM2017. - Submitted paper must be in the SAM2017. - Paper review has a conflict by PCMs.		
Main Flow:	1. PCC send a request to the PCM about the conflict review. 2. SAM2017 notifies the PCM about the review conflict. 3. PCM receives the notification about the conflicting reviews and the email addresses of the other PCMs. 4. PCM submits new review in the review submission page.		
Alternative Flows (1):	Not Applicable		
Post conditions:	PCC has been notified of the new review.		
Priority:	3		

Use Case ID:	UC10		
Use Case Name:	Set deadline		
Created By:	Wajdi Aljedaani	**Last Updated By:**	Team 6
Date Created:	09-24-2015	**Date Last Updated:**	12-06-2015
Overview:	Administrator shall be able to set up paper submission, review choice, review submission and author notification deadlines in the SAM2017.		
Actors:	Administrator		
Type:	Primary		
Preconditions:	- Administrator shall be logged in to the SAM2017.		
Main Flow:	1. The Administrator selects item he / she wishes to set the deadline for. 2. SAM2017 displays data about the item. 3. The Administrator enters the date value for the deadline. 4. The Administrator submits the form.		
Alternative Flows (1):	After Step (3) Administrator clicks on Cancel button. 1. SAM2017 redirects Administrator to home page.		
Post conditions:	Administrator has set deadlines successfully in the SAM2017.		
Post conditions alternative Flows (1):	Administrator has been cancelled to set up deadline successfully.		
Priority:	2		

Use Case ID:	UC11		
Use Case Name:	Create notification template		
Created By:	Muhammad Fazalul Rahman	**Last Updated By:**	Ian Shoenberger
Date Created:	09-24-2015	**Date Last Updated:**	12-05-2015
Overview:	Administrator shall be able to create templates for Notifications.		
Actors:	Administrator		
Type:	Primary		
Preconditions:	- Administrator must logged in to the SAM2017.		
Main Flow:	1. Administrator opens the create template page. 2. SAM2017 displays the create template form to the administrator. 3. Administrator enters the template for the desired notification that will be used in SAM2017. 4. Administrator confirms the template creation.		
Alternative Flows (1):	After Step (3) Administrator clicks on Cancel button. 1. SAM2017 redirects Administrator to home page.		
Post conditions:	Administrator has created template successfully and can be used by SAM2017 after the correlated event.		
Post conditions alternative Flows (1):	SAM2017 will not be able to use the new template.		
Priority:	2		

Use Case ID:	UC12		
Use Case Name:	Manage system		
Created By:	Ian Shoenberger	**Last Updated By:**	Team 6
Date Created:	09-24-2015	**Date Last Updated:**	12-05-2015
Overview:	Administrator shall have administrator privileges. The Administrator shall be able to update information about the registered users, paper submission and deadlines. Administrator shall also be able to delete registered users.		
Actors:	Administrator		
Type:	Primary		
Preconditions:	- Administrator shall be logged in to the SAM2017.		
Main Flow:	1. Administrator selects what he/she wants to update or delete. 2. Administrator clicks on the list of available users i.e. Author, PCC or PCM. 3. SAM2017 shows the list of available PCC, PCM or authors. 4. Admin selects the account he/she wants to update or delete. 5. Admin performs the action he/she wants to and clicks on submit button.		
Alternative Flows (1):	After step (2) if there is no users has registered in the SAM2017, SAM2017 returns an empty list.		
Alternative Flows (2):	After step (4) Administrator clicks on Cancel button. 1. SAM2017 redirects the Administrator to the main admin page.		
Post conditions:	SAM2017 has saved all the changes successfully that made by the Administrator.		
Post conditions alternative Flows (2):	SAM2017 has cancelled to update the changes successfully which made by Administrator.		
Priority:	3		

Use Case ID:	UC13		
Use Case Name:	Manage account		
Created By:	Wajdi Aljedaani	**Last Updated By:**	Team 6
Date Created:	09-24-2015	**Date Last Updated:**	12-05-2015
Overview:	Submitter/Author, PCC or PCM shall be able to update the personal information (first name, last name, email and password).		
Actors:	Submitter/Author, PCC or PCM		
Type:	Primary		
Preconditions:	- Submitter/Author, PCC or PCM shall be logged in to the SAM2017.		
Main Flow:	1. Submitter/Author, PCC or PCM navigate to edit profile page. 2. SAM2017 displays all the user information (first name, last name, email and password). 3. User changes the desired information that he/she wants to change and click on submit button.		
Alternative Flows (1):	After step (2) Submitter/Author, PCC or PCM clicks on Cancel button.		
Post conditions:	SAM2017 has saved all the changes successfully made by the Submitter/Author, PCC or PCM.		
Post conditions alternative Flows (1):	SAM2017 does not update any information and redirects the user to the main/home page.		
Priority:	3		

Use Case ID:	UC14		
Use Case Name:	Set reminder		
Created By:	Wajdi Aljedaani	**Last Updated By:**	Team 6
Date Created:	09-24-2015	**Date Last Updated:**	12-05-2015
Overview:	Administrator shall be able to set reminder for unassigned paper, review deadline and evaluating paper.		
Actors:	Administrator		
Type:	Primary		
Preconditions:	- Administrator shall be logged in to the SAM2017. - Administrator has created a notification.		
Main Flow:	1. Administrator chooses to create a deadline. 2. SAM2017 displays deadline form. 3. Administrator sets the date associated with the deadline. 4. Administrator links the notification related to the deadline. 5. Administrator chooses to submit. 6. SAM2017 displays a message to the Administrator that the deadline was created.		
Alternative Flows (1):	After step (4) Administrator clicks on Cancel button. 2. SAM2017 redirects the Administrator to the list of deadlines page.		
Post conditions:	SAM2017 has saved all the changes successfully that made by the Administrator.		
Post conditions alternative Flows (1):	SAM2017 will not display the reminder with details at one day intervals.		
Priority:	3		

Architecture model

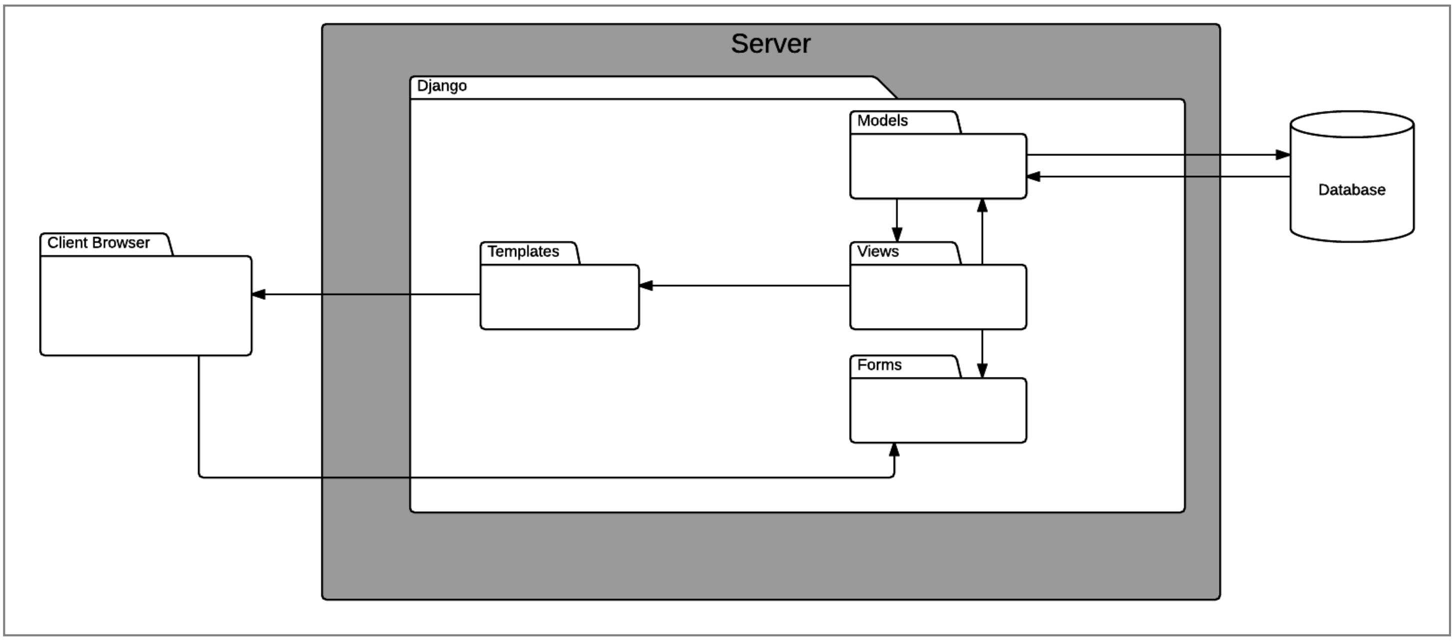

Figure 2 Architecture model diagram

Description of components

A Model is the single, definitive source of information about your data which contains the essential fields and behaviours of the data that you are storing. The model in Django is mapped to a database table. Usually, each model maps to a single database table and each attribute of the model represents a database field. The models in our application are located in **app/models.py** and the names of the classes are SAM2017User, Paper, Notification, NotificationTemplate, UserNotification, Deadline, PCMAssign, PCMPickList, PCMReview and PCCReview.

A View is simply a python function which takes the web request (HTTPRequest) and returns or turns it into a web response (HTTPResponse). The web response can be contents of a web page (HTML), redirect or 404 error. The Views in our application are located in **app/views.py** and the names of classes are RegisterUser, pcc_check, not_author_check, right_pcm, paper_author, ChangePassword, UpdateProfile, index, identify_user_group, user_login, user_logout, assign, hasThreePCMs, calc_letter_grade, calc_std_deviation, upload_paper, pcm_review, pcc_review, report, reupload_paper, download, get_note_from_template,spread_submitted_notification,spread_resolve_conflict_notification, spread_pcm_assign_notification, spread_report_ready_notification, handle_notification and pcm_pick_list.

A Template is a text document or normal Python string which is marking up using Django template language and template can contains template tags and variables. Django's template language is designed to feel comfortable to people who used to working or familiar with HTML and it is designed to strike a balance between power and ease. Django template variables gets replaced with values when the template is evaluated. Django template variables controls the logic of the template and allows us to transfer Python variables into HTML which makes it easier and faster to build dynamic websites. The templates in our application are located in **app/templates/app** and the names of HTML pages are base.html, assign.html, change_password.html, delete_this.html, download.html, index.html, login.html, pccrevire.html, pcmreview.html, register_user.html, report.html, reupload.html, update_profile.html, upload.html and wishlist.html.

A Database is designed for creating, editing and maintaining the records. Moreover, it allows user to store data in the form of structured fields, tables and columns, which can then be retrieved directly. In Django, there are different kinds of database that can be used, but in our application we used SQLite which works fine for small web projects, as we have. The database file in our application is located in **SAM2017/** and named as db.sqlite3.

An URL Dispatcher provides a simple way that can map URLs to view code using a simple pattern matching language. The patterns have number of URLs which are checked one by one and if one of the pattern match with a request, a particular view callable is invoked. The URLs file in our application is located in **app/urls.py** and the names of the URLs pattern that we have are register, chg, updateprofile, login, upload, reupload, assign, download, logout, handle_notification, wishlist, pcmreview, pcc_review and report.

A form file generates forms that can be used in HTML. The HTML form is the most common way to send data from a client application to a server. Also, it is responsible for taking some user input, validating it, and turning it into Python objects. Forms can resemble paper or database forms because web users fill out the forms using checkboxes, or text fields. The forms in our application are located in **app/forms.py** and the names of the forms are PCCReviewForm, PCMReviewForm, DocumentForm, ReuploadForm, UserForm, RegisterUser, ChangePasswordForm and UpdateProfileForm.

A Web Browser acts as the client side application and provides the UI generated through the views to the user. For the project, in addition to providing the basic functionalities to the user, the browser parses the Javascript to handle AJAX requests to different views. These views return JSON objects which are used to provide notifications to the users in real time. The information obtained through JSON is populated into HTML objects and the displayed in the UI. Most of the javascript code is present in **app/templates/app/base.html** as it is to be displayed in every page, except the login and registration pages.

Dataflow of the architecture

At this point we have a good understating of each individual components. The web browser is a place where you can test or display the project. In the browser, we have to type the URLs (urls.py) which is indicated in the diagram by URL dispatch. The URL dispatch maps the requested URL to a view (views.py) function and calls it. If caching is enabled, the view function can check to see if a cached version of the page exists and will return the cached version. It will then avoid all the further steps. When the view function checks for a cached page, it performs the request action which normally involves reading and writing to the database, but might also include other tasks as well. The model (models.py) defines the data in Python and interacts with it. Typically the models are contained in a relational database such as MySQL, SQLite, etc. However, other storage mechanisms are possible such as XML, text files, etc., but in our case we used SQLite.

The Template is used to return HTML pages. In Django template engine, all the power needed for the presentation logic is included. After performing any requested tasks, the view will return an HTTP response object to the web browser which generally happens after passing the data through a template. The view usually saves a version of the HTTP response object in a caching system (caching framework) for a specific period of time. However, you can clear out the caching framework by closing the web browser and reopening it again.

Domain Model

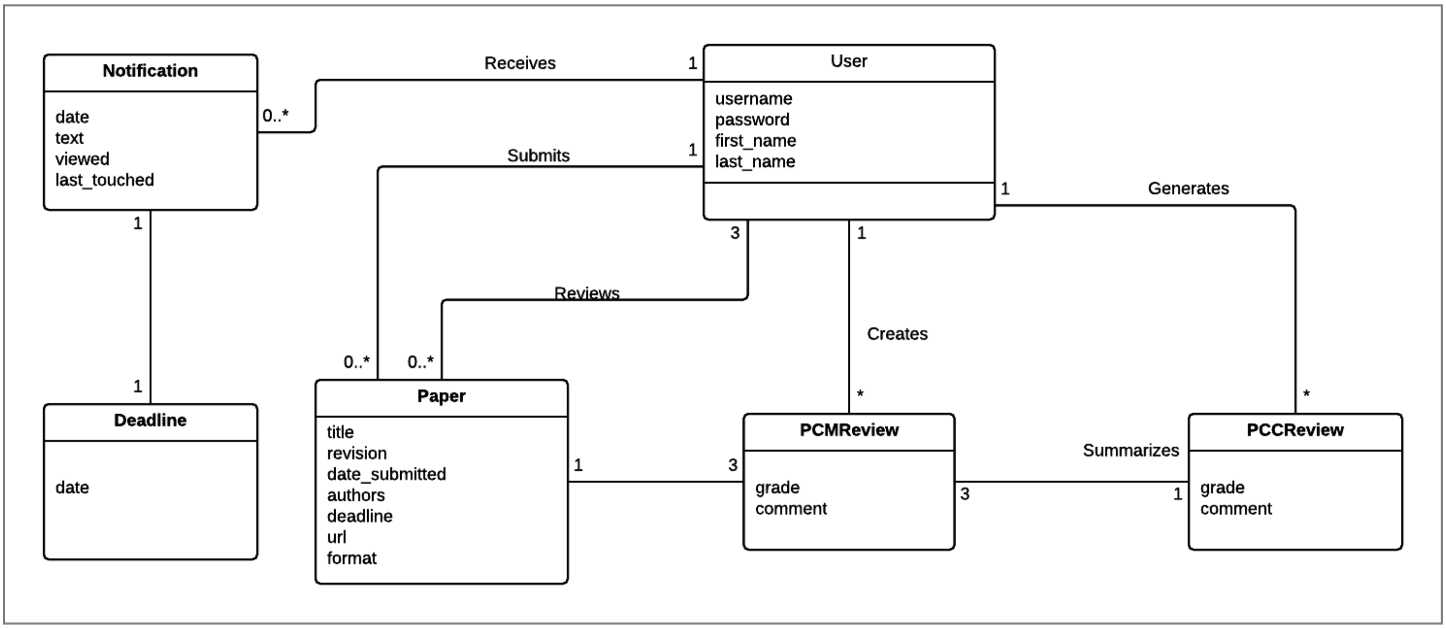

Figure 3 Domain model diagram

Design class diagram

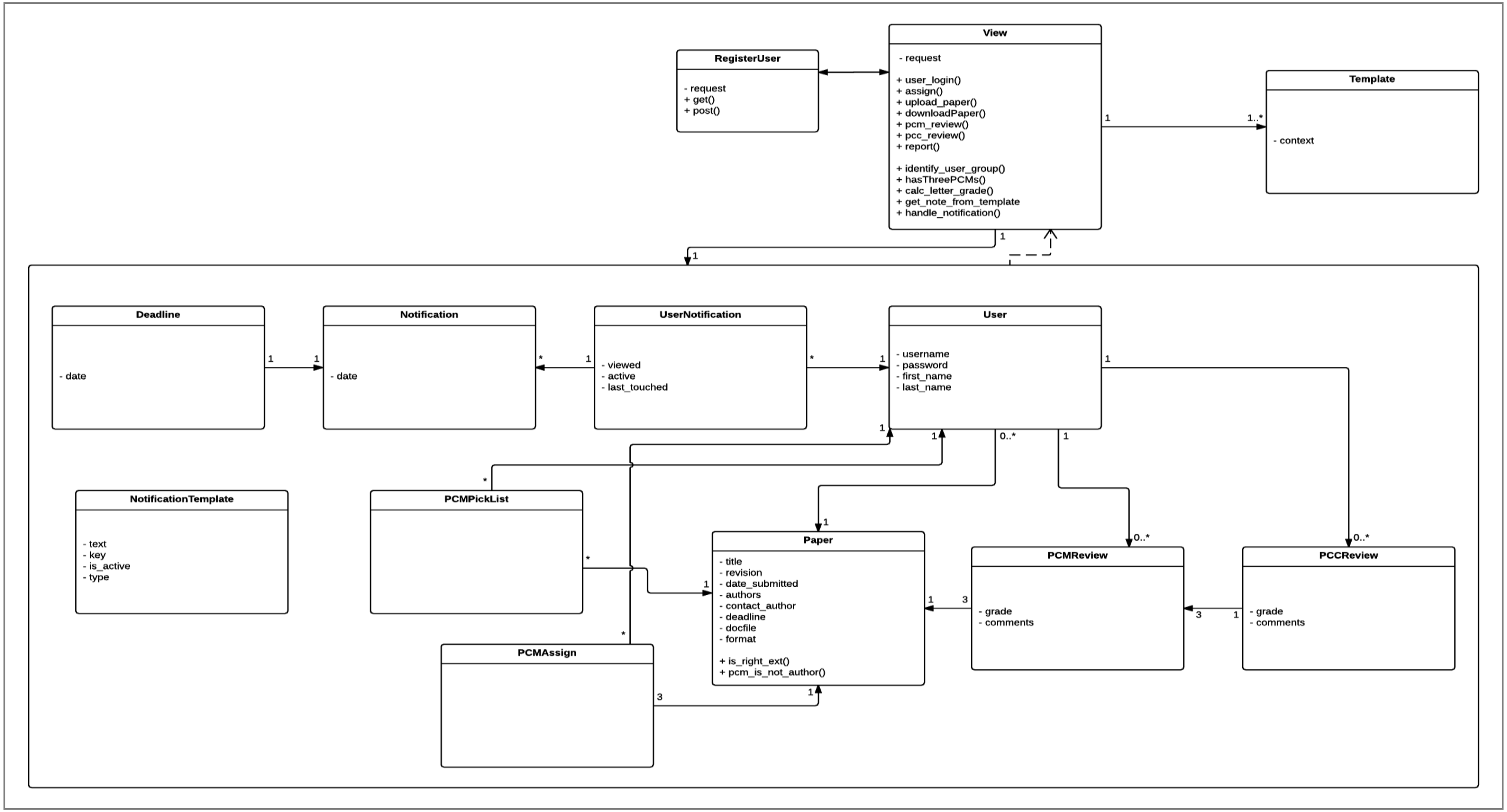

Figure 4 Design class diagram

System sequence diagram

Submit paper

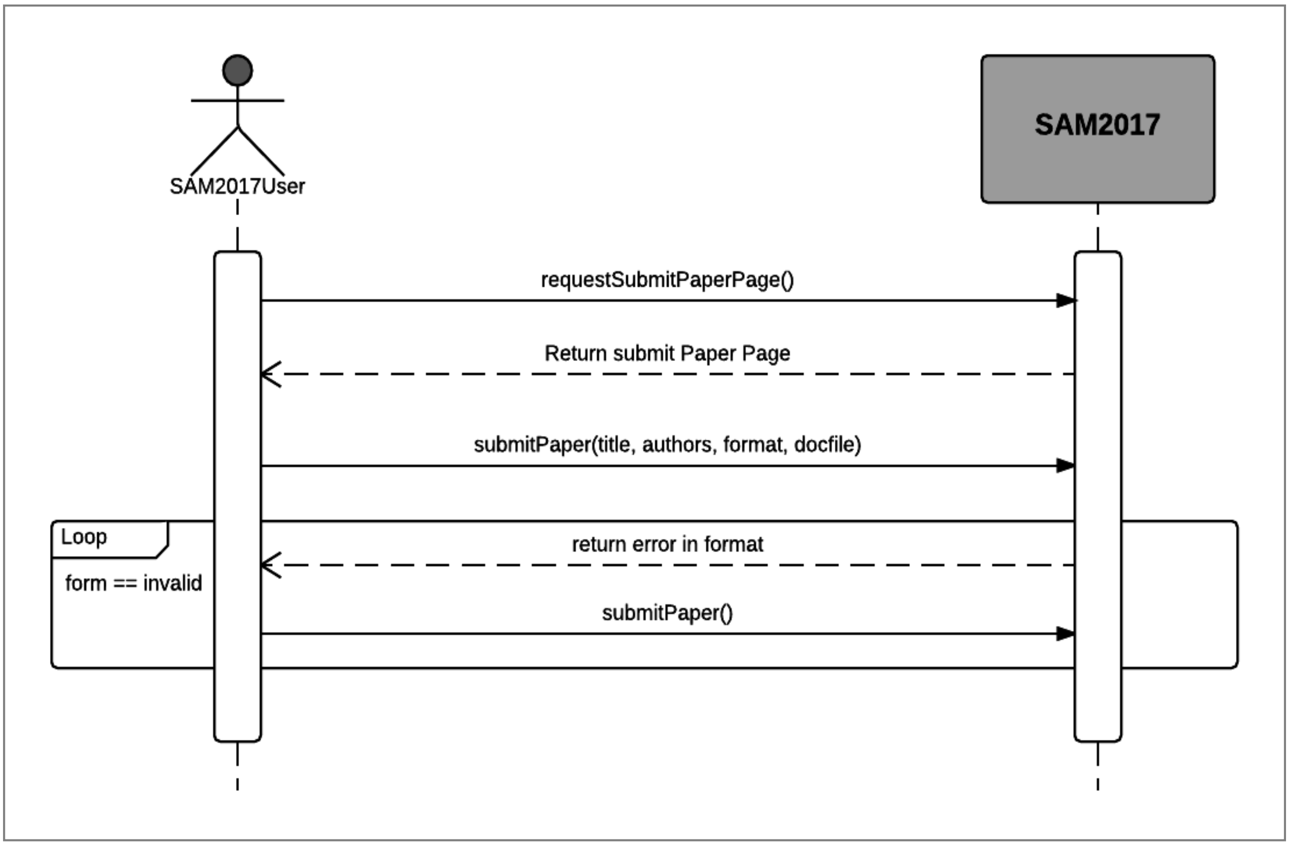

Figure 5 System sequence diagram (Submit paper)

Assign paper

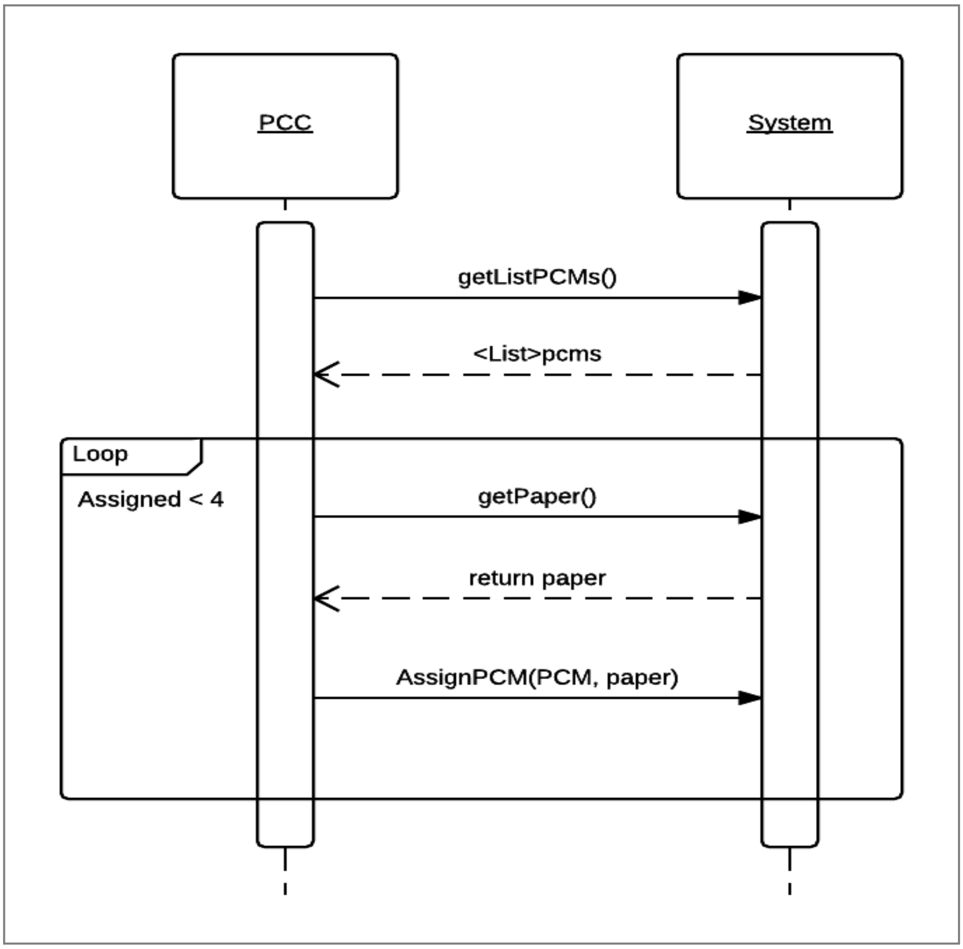

Figure 6 System sequence diagram (Assign paper)

Generate report

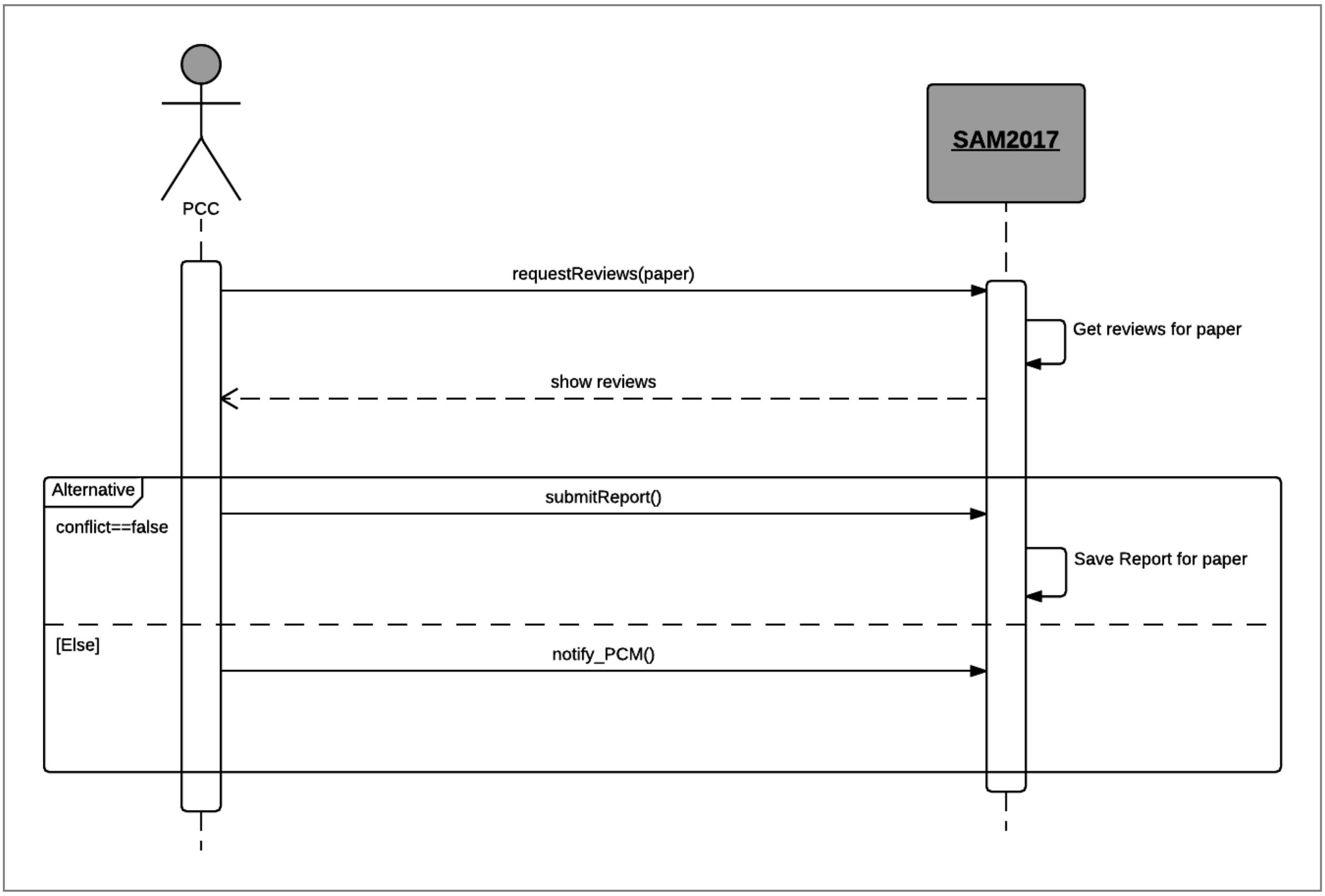

Figure 7 System sequence diagram (Generate report)

Behavioural diagram (Sequence diagram)

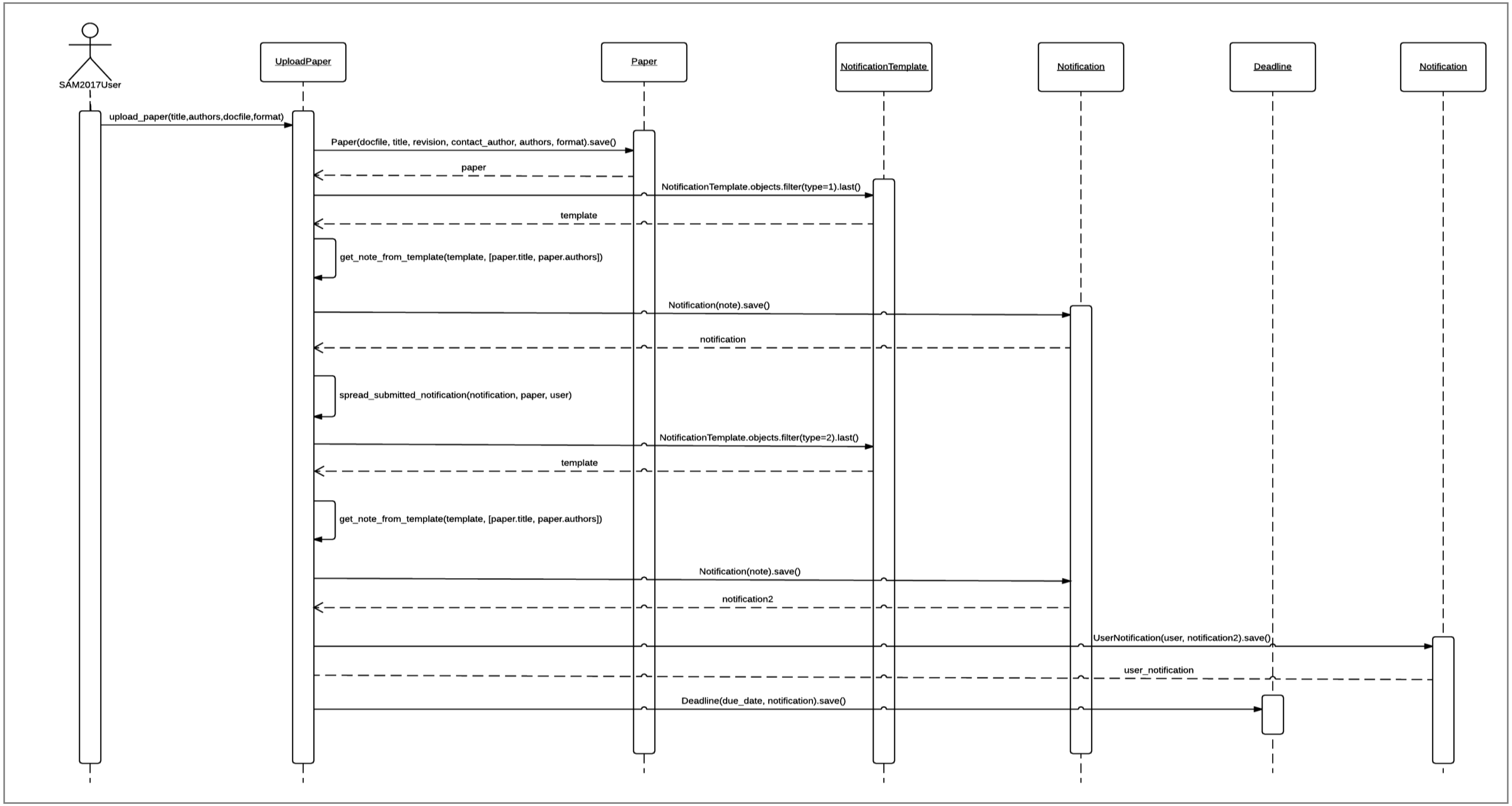
SAM2017User
UploadPaper
Paper
NotificationTemplate
Notification
Deadline
Notification
upload_paper(title,authors,docfile,format)
Paper(docfile, title, revision, contact_author, authors, format).save()
paper
NotificationTemplate.objects.filter(type=1).last()
template
get_note_from_template(template, [paper.title, paper.authors])
Notification(note).save()
notification
spread_submitted_notification(notification, paper, user)
NotificationTemplate.objects.filter(type=2).last()
template
get_note_from_template(template, [paper.title, paper.authors])
Notification(note).save()
notification2
UserNotification(user, notification2).save()
user_notification
Deadline(due_date, notification).save()

Figure 8 Sequence diagram for uploading a paper

State Machine diagram

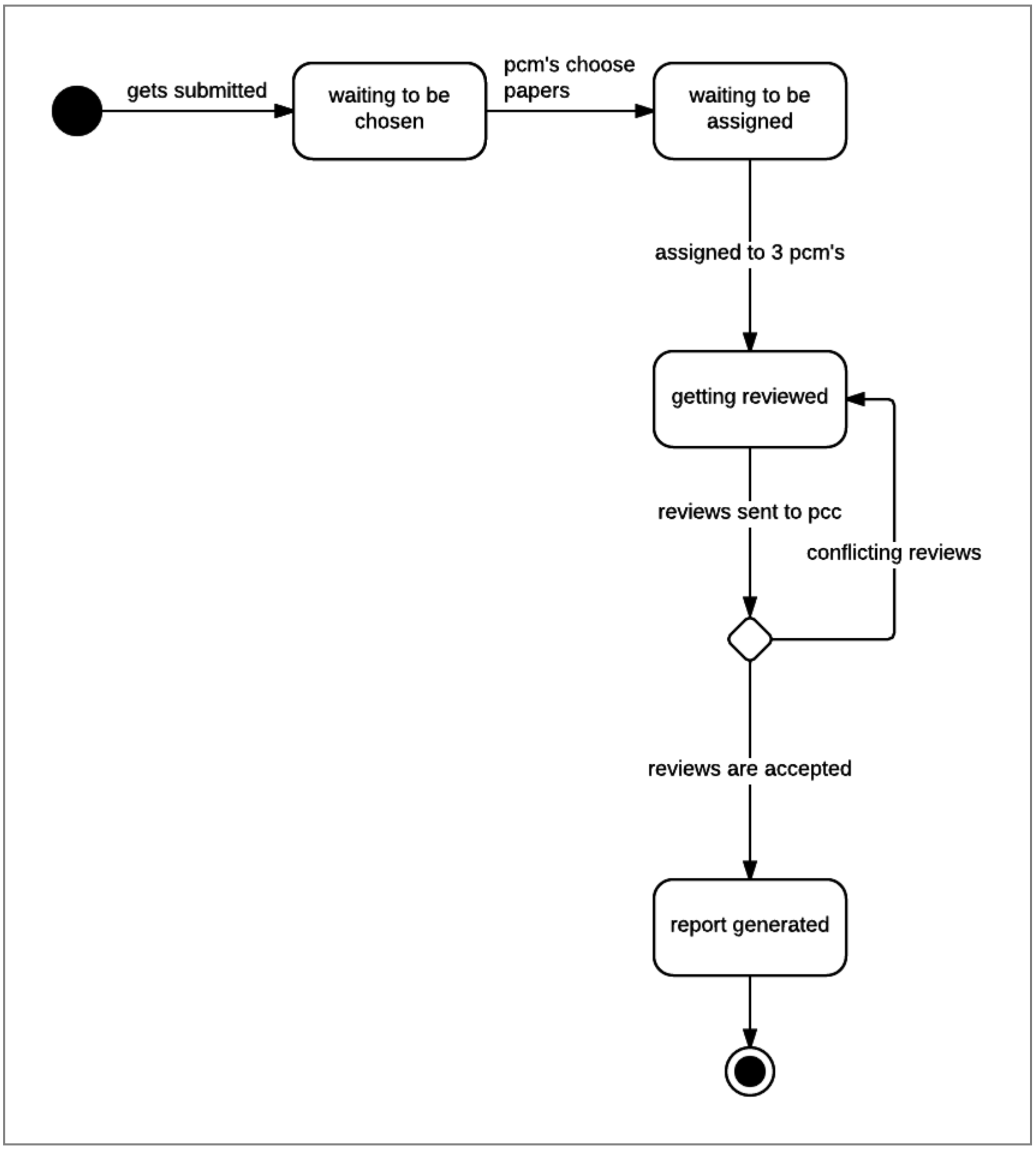

Figure 9 State diagram for paper