MARIST COLLEGE

To: Professor Pablo Rivas

Class: CMPT 475N Capping Project I & II

From: Liam Harwood
Cristian Hernandez
Daren Pagan
Juan Vasquez
Maxim Vitkin

Date: September 13th, 2017

Subject: Homework #1

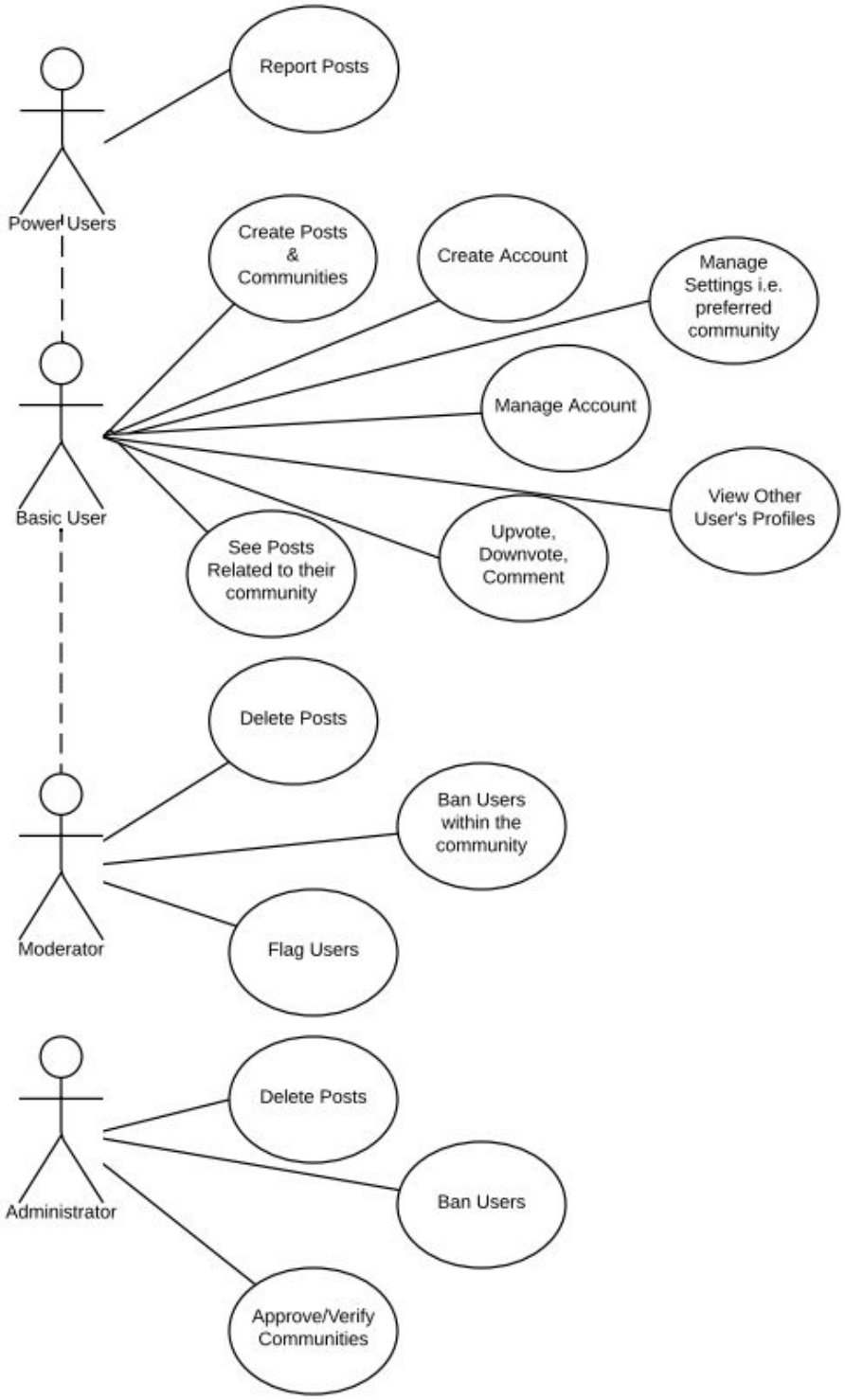
Date	Milestone	Tasks	Responsible
8/30/2017	Project Start	Form team, figure out roles and responsibilities. Schedule 1st meeting and establish plan for communication. Develop overall project plan.	All
9/5/2017	Pre-Client Interview Meeting	Meet and develop a list of questions to gather requirements from the client.	All
9/6/2017	Client Interview	Interview client in class to gather user requirements for PSN.	All
9/7/2017 -9/11/2017	Work on UML use case diagram	Create UML use cases on LucidChart.	Cristian
9/7/2017 -9/11/2017	Work on user requirement document and questionnaire	Compile and document user requirements and related questions asked to the client.	Juan, Daren
9/7/2017 -9/11/2017	Work on project plan	Work collaboratively to develop the overall project plan for the semester.	Liam, Max
9/12/2017	Homework #1 Final Check-in	Meet to check in on Homework 1 status and finalize plan.	All
9/13/2017	Homework #1 Due	Submit HW1, including project plan, user requirements, use cases, and interview questions.	All
9/14/2017 -9/18/2017	Work on Homework #2	Complete ER diagrams for the PSN database. Document as needed.	Cristian
9/14/2017 -9/18/2017	Research front-end and back-end technology stack	Research different options to use for our web technology stack and decide what to use.	Liam, Max
9/19/2017	Homework #2 Final Check-in	Meet to check in on Homework 2 status and finalize ER diagrams.	All
9/20/2017	Homework #2 Due	Submit HW2, including ER diagrams and all supporting documentation	All
9/21/2017 -9/25/2017	Work on Homework #3	Create initial mock-ups / wireframes for the UI.	Max, Juan
9/25/2017-	Determine Priority of Development	Order elements based on priority in order to determine the most important components for first prototype, second, etc.	Max, Juan, Liam
9/21/2017 -9/25/2017	Begin initial front-end development	Begin developing the user interface based on mock-ups.	Max, Juan

9/21/2017 -9/25/2017	Begin initial back-end development and database design	Begin designing the PSN database and developing preliminary back-end APIs.	Liam, Juan
9/26/2017	Homework #3 Final Check-in	Meet to check in on Homework 3 status and finalize mock-ups.	All
9/27/2017	Homework #3 Due	Submit mock-ups for the PSN user interface.	All
9/28/2017 -10/2/2017	Work on final draft of UML diagrams and project plan	Work collaboratively on the final version of UML diagrams and project plan	Cristian, Daren
9/28/2017 -10/2/2017	Continue developing front-end	Continue developing the front-end user interface for PSN.	Max, Juan
9/28/2017 -10/2/2017	Continue developing back-end	Continue developing back-end implementation for PSN.	Liam, Juan
10/3/2017	UML Diagrams and Project Plan Final Check-in	Meet to check in on and finalize UML diagrams and project plan.	All
10/4/2017	Final UML diagrams due, final project plan due	Submit final UML diagrams and project plan with input from instructor.	All
10/5/2017 -10/9/2017	Work on database design final version	Work collaboratively to finalize the database design for PSN.	All
10/10/2017	Database Design Final Check-in	Meet to check in on database design status and finalize draft version.	All
10/11/2017	Database Design Draft Due	Submit draft version of the database design for PSN	All
10/12/2017 -10/17/2017	Complete mid-semester peer reviews	Write first pass peer reviews for the group.	All
10/18/2017	Mid-semester peer reviews due	Submit initial peer reviews (first pass feedback).	All
10/19/2017 -10/23/2017	Work on IT requirements and network design	Write IT requirements documentation and design the network implementation for PSN.	Daren
10/24/2017	IT and Network Design Draft Final Check-in	Meet to check in on IT requirements and network design and finalize drafts.	All
10/25/2017	Draft IT requirements, including network design Due	Submit draft version of IT requirements and design of the network for PSN.	All

10/26/2017 -10/30/2017	Work on Homework #4	Develop database prototype.	Liam, Juan
10/31/2017	Homework #4 Final Check-in	Meet to check in on Homework 4 status and finalize the database prototype.	All
11/1/2017	Homework #4 Due	Submit database prototype.	All
11/2/2017 -11/6/2017	Complete front-end development for first demo	Make sure the front-end is ready for the first demo.	Max
11/2/2017 -11/6/2017	Complete back-end development for first demo	Make sure the back-end is ready for the first demo	Liam, Juan
11/7/2017	First Demo Final Check-in	Meet to check in on status of first demo and finalize anything as necessary.	All
11/8/2017	First Demo of Prototype	Complete first demo of overall PSN prototype.	All
11/9/2017 -12/13/2017	Complete front-end development	Finish front-end user interface for PSN in time for the final presentation.	Max, Juan
11/9/2017 -12/13/2017	Complete back-end development	Finish back-end development and database implementation in time for the final presentation.	Liam, Juan
11/15/2017	User validation and test plan finalized	Complete test plan for all aspects of the prototype. Document user validation tests to be performed by client.	All
11/22/2017	Client Visit	Complete updates to user validation (as required). Document one key question to ask the client.	All
11/28/2017	Draft Documentation Final Check-in	Meet to check in on status of draft documentation and finalize it for submission.	All
11/29/2017	Draft Documentation	Submit all documentation required for project in draft form,	All
11/30/2017 -12/4/2017	Complete Final Peer Evaluations	Write final peer evaluations for group members.	All
12/5/2017	Final Paper Check-in	Meet to check in on final status of documentation	All
12/6/2017	Paper Due - Final Documentation, Final Peer Evaluations	Submit all documentation required for the project in FINAL form. Complete final peer evaluation forms and submit.	All
12/12/2017	FINAL MEETING	Make sure everything is in working order for the final presentation.	All

12/13/2017	FINAL PRESENTATION	Present like Steve Jobs.	All

PSN Use Case Diagram



REQUIREMENTS

Basic User

- A basic user can create an account / profile page.
    - This account contains the user's full name, birthday, location, picture, and religions.
    - This account is not meant to be anonymous; users will use their real names in addition to a handle / username.
- A basic user can manage their account to change any information as needed.
- A basic user can create Posts (Prayer Requests) of up to 140 characters.
    - They can also add a description of up to 5000 characters, post updates of up to 140 characters, upload pictures associated with the prayer request, use hashtags to associate it with similar posts, and mark their own posts as complete (i.e. "Prayer Answered").
- A basic user can manage their settings, religions, and other preferences.
- A basic user can switch between different views such as least responses, most popular, or followed users timelines.
- A basic user can pray (similar to Facebook "like" or Reddit "upvote"), comment, etc. on posts.
- A basic user can downvote posts (posts are auto-reported with enough downvotes).
- A basic user can receive notifications for community activity or activity on their posts.
- A basic user can view profiles of any other users.

- A basic user can petition for a new “unverified” religion/community which is later approved.
- A basic user can join verified religions / communities.
- Users can earn points that contribute to their reputation rating on the site:
    - “Praying hand” points for praying for someone
    - “Halo” points for liking completed posts (answered prayers)
    - “Hammer” points for flagging reported posts
    - “Knee” points for upvotes on their own posts
- Users with high enough reputation gain Power User status.

Power User (includes Basic User Requirements)

- A power user can report problematic posts for moderators to examine.

Moderator (Creators of communities/religions, includes Basic User Requirements)

- Moderators can delete posts (in their communities).
- Moderators can ban users (in their communities).

Administrator (Developers / Site Owners)

- An administrator can delete any post.
- An administrator can ban users that have been flagged by THE SYSTEM.
- An administrator can approve and verify religions / communities.

Other Requirements

- Preferred technologies for the implementation are Linux and PostgreSQL.
- The web app should work in both Chrome and Safari browsers.
- A mobile app is necessary and the website should be mobile-ready.
- Each user should have a single encrypted password for authentication.
- Each user can connect account to Facebook, Twitter, or email.

CLIENT QUESTIONNAIRE

- Are you open to changing the name from Prayer Social Network to something else?
- What will this product accomplish that pre-existing social media outlets such as Facebook, Twitter, and Reddit don't already accomplish?
- How do you want to handle image storage?
- Are we only allowing people with a certain amount of reputation to report?
- What level of anonymity do you want users to have?
- Will there be a timeline that users can look through to see prayers from different communities, most popular, etc?
- Is the follow system a two-way connection like LinkedIn or a one-way connection like Twitter?
- In what situation would someone downvote a prayer request instead of reporting the user?
- Will non-registered users be able to view prayers?
- How should we handle user authentication?
- Can you create your own religion?
- Will there be moderators and/or administrators for the social network?
- Are there any preferred technologies for the implementation of the social network?
- What should be the aesthetic of the website? (i.e. What should it look like?)
- Should there be a super admin that has more power than the moderators?