Project Backlog

Project title: ReportIt
Team members: Juncheng Tang, Yingtong Chen, Liang Zhang, Yuting Guo, Yaoxi Liang

1. **Problem Statement:**

Public concerns appear in all around the world and should be seriously taken care of. However, properly managing and resolving those concerns can be a difficult task. Public service agencies need a convenient and well-managed platform to receive and respond to people's need, and the public needs an integrated platform to express their concern to different agencies.

2. **Background Information:**

Many residents in West Lafayette are facing various annoying problems which they want to report to corresponding departments. The reporting process is full of misunderstanding and frustration for both the residents and department officers since the resources support them this process is limited. Residents may have to waste their whole day in the department for a long time due to the lack of officers, and some of them are even not sure which department to report to. The department officers who receive tons of reports every day may mix reporters' information causing problems. All of our team members are residents in West Lafayette and some of us had experience of dealing with the time-consuming reporting process. Therefore, we aim to develop a web app that make this reporting process simple and organized to assist people who want to report case online with simple narration and clicks as well as department officers who want to organize report documents well online.

3. **Environment:**

We decide to build our front end interface with javascript, css, and html. We will be using external libraries such as Bootstrap, jQuery for UI-optimization. Python Django will be our primary tool to develop the backend. Our choice of database is Postgresql which will store all the reporter and agency information. We will also store the concern reports data in our database. We will use Amazon AWS S3 to store media data like images and videos attachment of the concern reports. Finally we will deploy our application to Heroku.

4. **Functional Requirement:**

Backlog ID	**Functional Requirement**	**Hours**	**Status**
01	As a user, I would like to sign up a reporter/agent account	**5**	**Sprint 1**
02	As a user, I would like to have a profile picture	**3**	**Sprint 1**
03	As a user, I would like to be able to change my password	**5**	**Sprint 1**
04	As a user, I would like to reset password when I forget my password	**2**	**Sprint 1**
05	As a user, I would like to view profile of other users.	**4**	**Sprint 2**
06	As a user, I would like to logout my account securely	**2**	**Sprint 1**
07	As a user, I would like to have my profile page	**3**	**Sprint 2**
08	As a user, I would like to view all the concerns	**4**	**Sprint 2**
09	As a user, I would like to change my profile page	**5**	**Sprint 2**
10	As a reporter, I would like to sign up with Facebook account	**4**	**Sprint 2**
11	As a reporter, I would like to sign up with Google+ account	**1**	**Sprint 2**
12	As a reporter, I would like to submit concern to one or multiple community agents	**10**	**Sprint 1**
13	As a reporter, I would like to upload image attachment to the agent	**2**	**Sprint 2**
14	As a reporter, I would like to view agent’s information	**3**	**Sprint 2**
15	As a reporter, I would like to search for any similar concerns	**15**	**Sprint 2**
16	As a reporter, I would like to upvote the concerns I like	**6**	**Sprint 2**
17	As a reporter, I would like to view all the concerns I sent	**3**	**Sprint 2**
18	As a reporter, I would like to delete my concern	**1**	**Sprint 1**

19	As a reporter, I would like to edit my concern	**5**	**Sprint 1**
20	As an agent, I would like to view all incoming concerns	**3**	**Sprint 2**
21	As an agent, I would like to respond to incoming concerns	**5**	**Sprint 2**
22	As an agent, I would like to be notified for new concerns	**6**	**Sprint 2**
23	As an agent, I would like to rank my incoming concerns	**4**	**Sprint 2**
24	As an agent, I would like to respond to concerns to reporters via email	**5**	**Sprint 2**
25	As an agent, I would like to redirect mis-directed concerns to other agents	**4**	**Sprint 2**
26	As an admin, I would like to verify agent’s information to avoid fake agent accounts.	**6**	**Sprint 1**
27	As an admin, I would like to delete spam concerns which have negative impact to the society	**4**	**Sprint 2**

5. Non-Functional Requirement:

Security:

Django is a matured python web framework development tool, which is secured enough to protect us from many attacks through website. As a result, we do not need to worry about problems like cross site scripting in our project.

However, we do need to worry about problems on inappropriate usage of our website. For example, we need to make sure that user does not maliciously perform certain operation for too many times and cause any traffic issue in our network server. We will be enforcing such rule with verification email and recaptchas on important operations.

It is also important for us to make sure that no malicious file is attached (and uploaded) to our system. We can enforce the security by limiting the executing privilege of the uploaded files. Since we will not be running any file on our server, it is necessary for us to remind our user of such potential problems.

Lastly, we need to help our user to protect their account by forcing them to set up a strong password.

Scalability:

Currently, our web app provides service for users in West Lafayette. In the future, we will be able to realize the same functionality to users in United States and even all over the world.

Reliability:

We understand the robustness of our web app is crucial for both reporters and agency. Since we will deploy our website on Heroku, host database on Postgresql and save attachments on AWS S3, we will be relying on those platforms for reliability concerns.

6. **User Cases:**

Case#1 - Sign Up	
Action	**System Response**
1. Click “Sign in”	2. Sign up page appears
3. Enter signup user information	
4. Click “submit”	5. “Verification email sent” Dialogue appears
6. Click “OK”	7. Redirect user back to login page

Case#2 - Make changes to profile such as password etc.	
Action	**System Response**
1. Click “My profile”	2.User profile page appears
3. Click “Edit”	4.“Edit profile page” appears
5. Make changes to the fields	
6. Click “submit”	7. Confirm dialog appears
8. Click “Confirm”	9. Redirect user to user profile page

Case#3 - Log in and log out	
Action	**System Response**
1. Click “Sign in”	2. Sign in page appears

3. Enter username and password	
4. Click submit	5. User redirect to profile page
6. Click logout	7. User redirect to login page

Case#4 - Submit concern	
Action	**System Response**
1. Click “Submit Concern”	2. Submit concern page appears
3. Enter concern	4. Display similar concern in drop-down window
5. Attach files (if necessary)	
6. Click submit	7. User redirected to profile page

Case#5 - View concern	
Action	**System Response**
1. Click “View Concern”	2. Concern Page appears with all the submitted information
3. For reporters, click on the button “Upvote Concern”	4. Pop up “Upvote Successful” dialogue
5. For Agents, click on the button “Respond Concern”	6. Redirect to the Respond Concern page
7. Click “Close” button	8. Close Concern Page and redirect back to the previous page

Case#6 - Respond to concern	
Action	**System Response**
1. Select concern to respond	
2. Click “Respond”	3. Display response page
4. Attach additional files (if necessary)	
5. Select target to send response	
6. Click submit	7. User redirected to profile page

Case#7 - View profile	
Action	**System Response**
1. Click on the name of the reporter/agent	2. Pop up new profile tag/window
3. Click on close profile button	4. Close Profile Page tag/window