Swachh Desi

The Service Learning App

**CAPSTONE PROJECT PROPOSAL**

**Group 9**

**Shiva Krishna K Avijit Nandi Sachin Gaurav Srivastava**

**TABLE OF CONTENTS**

* The Executive Summary
* The Project Background
* Aims and Objectives
* Relevance and Resources required
* Extensions and Risks
* Problem definition
* Primary Use-cases
* Risk list
* **The Executive Summary**

The Objective of this project is to provide a Service-learning(Wikipedia defines **Service-learning** as an educational approach that combines learning objectives with community service in order to provide a pragmatic, progressive learning experience while meeting societal needs) application to Indians who are interested in keeping their city clean.

In our day to day life we find several places in our communities which are unorganized/unclean, hence they require special attention from the government. But in reality our government may not be able to address all the issues/concerns all by itself, hence our idea is to build an application where people own these problems and try to find solutions to these generic problems.

* **The Project Background**

This application offers service to two parties,

1. Swachh Citizen - In our use case these are the reporters of a problem. If any responsible citizen finds any place not clean whether it can be a place, building, government property like railway station..etc. can report the problem by creating a new incident with details like description of the problem, place of the problem, directions to the place, pictures of the place, number of people the service needs to fix it and the cost it may involve. Additionally they can fix a date to resolve the issue and list down the details of tools/resources needed to fix the problem. Once the problem is solved they may want to upload new photos after providing the service and close the incident.

2. Swachh Volunteer - These are the responders to a reported problem. Any interested/registered volunteers can see current Incidents by entering the location and see the details of it. If they are interested they can sign up for it and provide what kind of help they can provide to fix it.

* **Aims and Objectives**

1. We are planning on having a registered account for each user to create and respond to an incident.

2. To register an account there will be few mandatory fields, without which account creation is not possible.

3. Any registered user can create or respond to an incident.

4. Creator of the incident must provide enough details about the incident,

A. At least one picture of the problem,

B. description of the problem,

C. directions to the specified place

D. Proposed solution to the problem

E. Resources/Tools needed to fix the problem

F. Number of people need to fix the problem

G. Total amount to fix the problem

H. Total time to fix the problem

I. Proposed future date and time to fix the problem

5. Any registered user should be able to get a list of all the incidents(and their details) of a specified area

6. Any registered user can respond to any incident by signing up for it, and provide the list of resources/tools he/she can bring with them.

7. Once the incident is resolved, the creator can upload the latest picture and resolve the incident.

8. Any registered user can always go and check the list of incidents created and responded to.

* **Relevance and Resources required**

This project is relevant to our certificate programme because

1. It involves building a web application to work on both web browser and mobile browser. We learnt web application development in our course and we are planning on using ReactJS to build it.

2. It involves building a back-end REST API to respond to facilitate the web application. We are planning on using the Django framework for it as we learned the framework as part of this course.

3. It involves creating DBs to store the details of the incidents in a database, we are going to use a database which we learnt in the programme.

4. We are going to create different micro services of each independent module(web app, rest api server..etc) we are going to create for this project as we learnt micro-services in this programme.

5. We are going to implement CI CD pipelines for this project as we have learnt in this programme.

6. We are going to use Dockers and K8s to deploy our entire application.

* **Extensions and Risks**

1. Amount Donation - At the moment we may not be able to allow donations from unregistered users or responders, but we will try to consider it if time permits.

2. Star Ratings - We might bring the star rating of the users to each registered user based on their individual activity.

3. Content Moderation - Currently we are planning to allow only creators to moderate the content but if time permits we will allow moderator accounts to restrict few actions.

* **Problem definition**

Currently in our understanding there is no such platform exists in the market, and we are trying to solve this problem by providing basic functionality to Indians who are really interested in keeping the streets clean. We can consider social media platforms like Facebook, Twitter, Instagram as existing platforms to report issues publicly and there is not actually a streamlined process to address these issues. All social media websites provide a bare minimum communication channel to people.

With our approach we are trying to address this problem by provide the web app by implementing this as a main use case so that the concept does not deviate from what we intended to do. When people visit the website they either want to report an issue, or want to know what are active issues in their area, or donate for interesting activities(Donations/Money transactions are not actively being considered for this project as it involves heavy coding with limited timelines).

Below are some of the key points considered while designing the use cases

1. The target audience any registered user, they can browse through issues or respond to the issues
2. There is no ROI expected on this platform but if the platform grows big enough we can add third party ads and engage local stores for advertising the resources needed to finish the reported problems. For example, if the problem reported was a small wall then local vendors can supplies cement, and other utils for a lump sum amount.
3. Functional requirements of the system are that the app should be an web app and it should be usable on all the platforms like desktop browser and mobile browser.
4. Non functional requirements are actors in this use case. People drive the platform, but not any governing body. People actively participate in the system voluntarily and complete the pending tasks.
5. Key metrics or milestones of this project are, 1. people should be able to register into the systems and search for active issues, 2. They should be able to report a problem, 3. Other people should be able to respond to the problem, 4. Reporters should be able to close the problem with details of performed activity.
6. Artifacts in this project are a web app, which has a ReactJS front end, and a Python back end with appropriate DB to save details of the users and issues, and finally a deployment model which can scale up and down based on the load.
7. The process life cycle we follow for this project is we start with the design first, we will define all the dependencies, tools, technologies needed for this project. Once all the design, user cases and tech stack is finalized, we will start with the REST API definition, then we will use these API definitions as basis for both frontend and backend development which can run in parallel. And during this development process we can plan for the deployment strategy at the 2nd half of the project.

* **Primary Use-cases**

For this project to have a GO, we need to construct a few primary use cases so that they provide the flow of the project. These use cases should provide a skeleton to the project which can help team members to work on the respective items independently. With this understanding we have identified below 4 primary use cases.

1. Users should be able to register and login to the portal - We want to restrict the platform to only restricted users and otherwise privacy can be a big problem. Every registering user should provide details like their name, phone number, email id, date of birth etc. These personal details will not be visible publicly. These personal details will be visible to only Admins. Once anyone registers into the platform they can search for the active issues by a place, and browse through details.
2. Any registered user can create an issue by specifying details like place of the issue, direction to the place, type of issue, picture of the issue, number of people needed to fix the problem, date of resolution, and resources needed..etc.
3. Any responsible user can acknowledge an interesting issue by counting them in, they can also specify what resources/utils they can bring with them. Once the maximum number of people needed is reached no one can sign up for it. Same goes with resources as well.
4. On the resolution day everybody gathers and fixes the problem and creator provides the details of the resolution and closes the issue. Any registered user can go back and check the issue history they participated in.

* **Risk list**

Apart from the above mentioned use case the system can be expanded into other features as well, as there is a very huge scope for improvement and make it a big ecosystem. Below are the improvements we are thinking of as expansion but are not actively being looked towards for the completion of the capstone project.

1. Adding an Admin role for moderation - For this kind of a system there should be an Admin to moderate and actively approve newly created issues, approve new registered users..etc.
2. Addition of Donations - For some people participating directly may not be the best option but they want to donate some money to resolve the issue, hence adding a donation feature will be very good. But this involves lots of changes and a payment gateway hence we are not going with this feature.
3. We may not be able to provide an option to upload more than 1 picture of the issue as it may involve heavy UI changes for the web app. For any issue the creator can upload one image explaining the issue, and one more image after resolving the issue.