Shreenath | Systers 2017 GSOC Student Application

Community Questions

1. [Optional] Are you a Syster (www.systers.org)? Would you join if you are accepted?

Ans: I am not a *syster*, although I am a part of the dev community on slack. If given chance, I would definitely want to be a part of such an awesome community! :)

---

2. How can we reach you (email, GTalk, Slack etc.) if we have questions about your application?

Ans: I will be available via Slack (**@####**) and Gmail (**####@####.com**) regarding any questions about my application.

---

3. What is your github username(s):

Ans: ***####***

Project Specific Questions

4. Which Systers GSoC project are you applying for:

Ans: I am interested to work on ***HopperApp*** *- an Android application to help manage the Hoppers (Volunteers) at the Grace Hopper Celebration*

4a. What do you plan to accomplish over this summer for this project? (Should be in timeline format. Please tell us what project you want to work on, how you will approach working on that project, and what your milestones are with dates included. You may ask for help from mentors in Slack in the #gsoc-questions channel if you are unsure about this process. Although, being able to realistically estimate how much you will be able to accomplish is an ***important part*** *of this proposal.)*

Ans: I want to complete the Hoppers Android App, with all the features that I have mentioned in the Application over this summer for this project.

Note: Please go through the Prototype, App features and Technical Implementation first so that the proposed timeline becomes clearer.

Duration	Description
May 5 - 30 2017	**Community Bonding :** 1. Get acquainted with the community. 2. Get to know about the general practises and coding guidelines that need to be followed. 3. Go through the final implementation with the mentors. Any prioritising/deprioritizing will be done, if required.
May 30 - June 5 2017	Prepare a skeleton of the application corresponding to my proposal with dummy sessions and conference data, such that it works exactly as in the prototype :-)

June 6 - 12 2017	1. Finalize an API contract / data model for communication between server and the front end. 2. Setup the server with essential table schemas required for populating Hoppers App, such as - the task-table, which maintains all the assigned tasks, with the data structure defined below. The Hoppers table, which contains essential details about the hoppers, with schema defined in the App description. 3. Integrate and use etouches API to populate the sessions data for the conference, since it is how it will be done in the conference_App.
June 13 - 19 2017	1. Using Dropwizard framework, create REST API methods and expose them so that they can be used by the application, such as *getTaskListForUser*, etc. 2. Modify Android List activities to fetch data from given APIs for populating the data using Retrofit, which wholeheartedly supports android development.**[link]** 3. Now that data has been populated in the database, I will work on user authentication (Login Activity). This can either by done by etouches eReg, or at our end itself, by matching the hashes of stored credentials and the password entered by the user.
June 20 - 26 2017	1. Work on Offline Notifications which notify a user 5-10 mins prior to task start time. These will be done using android's AlarmManager to schedule notifications. 2. Also populate the **My Profile** screen in the application. 3. Work on the Maps integration in the Application to power Navigation to Sessions Venue. A simple implementation would be use the Geo coordinates provided by etouches API, and use the `action_view` intent to open the Maps Application for navigation.
June 26 - 30 2017	**First Evaluations** : Progress - 1. Application supports user login, and is able to connect to external server via given APIs, to correctly display the list of tasks in the given Sessions-tasks model. 2. The backend server contains the hoppers registration data, and serves APIs that populate android app. 3. **Offline notifications** few minutes before the next task. Todo - 1. Will create a Firebase Project, and integrate its SDK into our app for receiving **Online Notifications**. 2. An additional API will also be created to dynamically store Firebase registration tokens of the volunteers, whenever they get registered. This will be helpful in sending Task Relinquish/Swapping request notification to Hoppers via firebase. We can also customize the token to deduplicate the notifications by using Volunteers userId, if required.
June 27 - July 3 2017 **July 4 - 10 2017**	Work on the task relinquishment feature - 1. App is able to send relinquish request to the server in the form of a GET call to the `relinquishTask` API, containing the taskId to initiate the swap process. 2. Our Java application will then query the task database for *available* volunteers using a simple sql query, and send the push notifications to

	those selected volunteers. 3. We'll implement this by using FCM HTTP Connection Server. Our server application will use stored registration token to send notifications to the selected Hoppers by issuing an appropriate POST request to the Google Provided servers. 4. When another hopper accepts the request, our client will hit the application server via an `acceptRelinquishTask` API for the given task, after which the task gets updated with the new volunteer, and both, the requester and the accepting hopper get a corresponding notification. Due the way this is implemented, any other
July 11 - 17 2017	1. Work on calculation of total shift time of volunteers per day. Every time there is an app refresh or a swap request, we will calculate total time by adding all the task's individual duration. If it is below 8 hrs, a **RED** bar will be displayed. (Pls see the prototype or [link to the description]) 2. Similar functionality will be implemented server-side, maintaining a list of Under-contribution volunteers for the moderators. We can build a simple bootstrap dashboard monitoring all the volunteers.
July 18 - 24 2017	1. Integrate the **pub-nub-android** chat client in the app based on this repository's implementation
July 24 - 28 2017	**Second Evaluations** : By this time - 1. Volunteers can swap tasks. 2. Moderators and Volunteers can view the 8 hour duty rule. 3. Volunteers will be able to chat among themselves.
July 25 - 31 2017	1. Add similar feature to the backend, so that moderators can also talk in the similar chatroom.
Aug 1 - 7 2017	*(This period will be little busy since I'll be shifting. I have discussed about it later in the proposal. Hence added lightweight features for this week.)* 1. Add Announcements feature on the server side, to allow moderators to send announcements to all the volunteers. We can either use the firecloud console, or create a simple API on our application server that makes an appropriate POST request to Google Provided servers, as described above. 2. The Announcement Notification handling will be added and maintained in the android app.
Aug 8 - 14 2017	1. Add the **Contact Moderators** button. The hopper will directly be able to call moderators by click on the phone number. This will be implemented using Action.Dial Intent. 2. Add the **Updates** functionality. a. Server sends an 'update' Notification to all the hoppers' apps which is received and shown in the side menu. b. When user on the app clicks on it, we basically reload all the tasks and data from the server. 3. Allow users to Edit their Contact details in the My Profile section. This will correspondingly update the database. This can be done by issuing PUT call with the new user data to our application server.

Aug 15 - 21 2017	1. To implement this feature, we will just need to maintain an additional variable in systemPrefs that maintains the current conferenceId. Since our data model attaches a conference with each Task, just changing the current conference and repopulating the task list would work. 2. Add the **Future Conferences** Button on the side menu. From the list of all active conferences, choosing any of the conferences just changes the current conference and accordingly the task list is updated. 3. Will work on bug fixes and final testing of the application.
Aug 21-29 and onwards	1. Bug fixes and code sanity.

I am well informed about the commitment in terms of work hours required, and I am comfortable in dedicating 8 hours everyday for development, during the GSoC period. Since I will be staying at home for most of the period, I will have no other engagements, and will be able to connect with Mentors even at odd hours. Having completed on our Company's conference App (details in later sections) in three weeks, I am sure that I will be able to manage the proposed timeline for development of the Hoppers App..

---

5. Please answer questions appropriate to the stage of the project/your contribution level:

5b. [Optional] Include wireframes of the new applications. Which prototyping tool did you choose? Include notes of each element (i.e. accordion menu, bottom navigation, etc.) and why you chose that method/icon.

I used MarvelApp as my prototyping tool. I decided to use this, since it gives a very intuitive prototyping, which makes it very easy to visualize how the app is finally gonna look like.
How to use the tool : Click anywhere on mobile to highlight where the clickable areas are. From then on, please navigate as desired. I have explained the features in the section below.
Prototype : **https://marvelapp.com/34c8275**

App Features

5c. [Optional] For new apps, what do you believe are the minimum features required? Why? For established apps, what areas do you believe can be improved? How?

Note : Please visit the Technical Details/Implementation section first to understand the justification behind my approach. After going through that, my usage of respective APIs and functions will become clearer. Thanks! :)
Ans: For my first design of the Hoppers App, I believe following features are a must have to help volunteers manage conference effectively.

1. **Login** [link] **-**
   Essentially, volunteers will have to login to see the assigned duties. Since volunteers would already have had registered via a separate mechanism, we would just query the user database with the provided credentials.

2. **My Schedule** [link] - This should be the first thing any volunteer should see after logging in the app. From the APIs that are provided by the hoppers' backend server, we'll fetch all the tasks allocated to the volunteer at once, using an `getTasksListForUser(userId)` API. Every task element is of the type `Task` defined in the data model. Each task, will then be grouped on the basis of its Sessions, and then populated in the App.
   From the Front-end perspective -

   a. For each hopper, the list contains the sessions he has to manage for the given conference. [link]
   b. Sessions List items are expandable, which initially show only session title and the time interval of the session. When clicked, it expands to show the list of tasks that need to completed for that particular session. [link]
   c. The list of tasks for the session is also expandable, where each task list item expands to show the description of the task to be performed. [link]
   d. The task description also includes a Maps Link to **respective venue of the task**. This can be implemented using Google Maps API, where we can create custom Markers for all the session locations, and navigate from one place to the other. Since our app is focussed on Managing tasks, I have not focussed deep on the implementation. Also, since Conference App will include a more complex navigation portion for the conference, we can work together to integrate some of the complicated the same feature to help the volunteers.
   e. The description box of the task list item also contains three buttons, which are discussed in points 4, 5 and 6. [link]

3. **All Sessions** [link]**-** Since volunteers will also be attending other sessions of the conference when they are not managing, a conference schedule is a necessary element in the application. With the help of a tabbed design, the volunteer can just swipe to have a look at the conference schedule. This way, volunteers need not download a separate application for viewing the conference schedule. We can fetch these details from the etouches API the same way it would be fetched for populating the conApp. We can use the same Maps integration that is used in the My Sessions tab, since it's eventually the same map..

4. ***A Relinquish request*** [link]**-**
   Simply put, if a volunteer finds him/herself unable to complete the task by themselves, they should be able to request someone else to do the same on their behalf. For our purposes, since we maintain a record of all the volunteers' *schedule,* we can easily send a notification to all the volunteers who are free in the required time slot. Frankly, this would be like sending an uber request, all the nearby and free volunteers will get a notification, whoever accepts the first, will have the task added in their schedule.
   a. Additionally, Other users will get a notification of the Task Request in a separate section. They can view the request, accept it, or simply ignore it.

   b. Apart from the **Requests** section of the App, all such tasks are updated on one of the moderators' dashboards, where they can manage the event. If they feel like any of the task hasn't been picked up by a volunteer, they can manually assign it to someone.

   c. **What happens if a volunteer does not complete his mandated 8 hours, by relinquishing his tasks, etc.?**

The App constantly calculates the total duration of the volunteering work based upon the duration of the task that he is assigned to do. When a volunteer, by any means, (eg. by relinquishing tasks) gets below the 8hrs threshold, a RED BAR is shown on the top notifying the volunteer of this, and directing him to contact the admins regarding that. The admins can, then allocate him any other task required to complete his contribution.

d. Whenever a user presses a relinquish button for a task, an HTTP GET request is sent to our server with the taskId, which changes the task's status to ***FOR_SWAPPING***, and updates the task into the moderators' dashboard. At the same time, we query for all the volunteers who do not have any assignment during that time, then send a notification to all of them, which can be seen in the their ***Requests*** side-menu item. When a volunteer accepts the request, an HTTP GET request is again sent, and the task's status is changed to normal,its assignee is changed in the backend, the task gets moved from user1 to user2, and it is removed from the dashboard. At this point, we'll also send an update notification to all the other volunteers that will be handled such that the deprecated swap request is removed from their app.

e. ***What happens if someone presses the accept button once the task is already accepted by someone else?***
For processing the accept request, the server checks if the task is still in *FOR_SWAPPING* state. Without this status check, that server doesn't allow anyone to accept it. This way we ensure that task in accepted only once.

5. ***Discussion Room*** [link]**-** This is a must-have requirement for the app, since this will serve following purposes :
   a. It will allow important information broadcasts to the volunteers at once.
   b. It will also allow discussion between volunteers and ask for guidance amongst each other.
   c. Since Conferences work in fashion of several sessions being run at the same time slot, all such volunteers that are at the same place at the same time (all the volunteers that share the same duty time at a place), will be automatically added in the chat server, where moderators can directly talk to them, and volunteers can discuss amongst themselves. We'll be following this repository for adding a chat functionality using **pubnub chat server**, which provides the flexibility that we can control (via code) what chat room a person is connected to. This way, after the session is over, we can disconnect the chat, and switch to a new chat room for the next session, with different volunteers.

6. **Mark as Done** [link] **-** Once a volunteer completes the task, he should be able to update the status of the task as complete. This will help in coordination and regular updation of the central management team. If required, we can have the data analysed to do a better assignment next time. A Simple GET request to `markDone API`, with the taskId will prompt the server to change the task status to DONE. Same will be done at the application end.

7. **Android Notifications** [link] - We should notify the volunteers of their upcoming tasks beforehand, and hence notifications would be necessary. We'll be using Firebase Cloud Messaging to send Online Notifications from the application server. The implementation is discussed in the Timeline.

8. **Future Events** [link] - As we can see in the prototype, **`Systers MegaConf`** is the current/selected conference. For conferences/events who are served/managed from the same API set, a volunteer will be able see his/her allocation and schedule for the next conferences, just like he's able to see for the current conference where he is volunteering.

9. **Contact the supervisors** [link] **-** This is also a must-have functionality in the application, in case a volunteer faces some trouble, they can easily contact the management of the the conference/event.

10. **Updates** [link]- In case one of the volunteers couldn't turn up the venue, then those tasks would need to be reallocated. In such cases, after organisers update on the server they'll be sending a notification, and all such updates will be shown in the sidebar options. After clicking the option, the app will sync with the server fetching the refreshed data from the server.

11. **Announcements** [link] - Moderators can communicate with the volunteers from chat room too, but that 1) that is a temporary chat room, 2) only active volunteers will be in that chat room. Hence, if the moderators need to communicate to all the volunteers at once, they can choose to send an Announcement/Broadcast message to all the volunteers, which will be displayed in the Announcements section of the app. Implementation is discussed in the Timeline.

12. **My Profile** [link]- This page of the android app is to give a little touch of personalization to the application. Apart from this, it contains all the ways in which a volunteer can be contacted if required.

13. **Signup** - Since this app is focused on helping volunteers view and complete their assigned tasks for an event, ***registration & task allocation do not need to be integrated in the hoppers App.*** *Few reasons for doing this are -*
    a. *Since volunteers do not get registered at once, it would be very inefficient to assign tasks to them on first come first allocation basis. For this, we should just ask all willing volunteers to fill in a form submitting all their availability details (place, time, expertise, etc.), and run an allocation algorithm on a particular date and time,* ***once****. After this allocation, all the volunteers are notified of their assignment.*
    b. *Since allocation is supposed to be a one time task and would require the data of all registered applicants at once, it would be more efficient to be run it on a single server, and expose the allocation results via an api, per volunteer.*

    Hence, I haven't put the register button on the application login page. If required, a register button can be put which links to an online form to complete the signup.

Technical Details/Implementation -

Data Model/API/ Communication between Server and App -

Data Model : I propose a Task based data model. Since a volunteer is assigned to handle complete sessions and each session can have multiple tasks, we can represent the whole arrangement with the help of Task Entities. As we can see in the data model below, we will easily be able to group tasks based and their sessionId. Hence, each session will constitute several Task Entities. Following are basic entities (pseudo code) -

Task { taskId (integer), taskName, taskDescription, conferenceId, sessionId (integer), sessionName, taskStartTime, taskEndTime, assigneeId/volunteerId, taskStatus }	**Volunteer/hopper/Assignee { userId, userName, contactDetails(list of strings), Encrypted_credentials }**

Communication : There are several approaches that can be used for communication between Hoppers App and the central server at where the data related to volunteer registrations is kept.

1. **[Discussion][not preferred] etouches**
   An approach that looks very easy at first, is using the etouches' eSocial [link] API for managing sessions and eReg API [link] for user registration and user data management, since etouches already caters to our conApp. However on deeper inspection with respect to our specifications, etouches API falls short. Following is my analysis of etouches' API.
   a. One obvious benefit is that we wouldn't need to maintain a separate user database at our end, and a web interface wouldn't need to be developed for the moderators, since a bare minimum ui is provided by the etouches framework. **However**, that would be a little hard to get used to, since the API is ***not*** designed in form the of Tasks Assignment and Management Format.

   b. eSocial has a concept of meetings, which can be used to emulate our task entity. Each meeting in etouches' data model contains a list of attendees, which can act as volunteers for that task. We can create tasks with assigned volunteers (PUT createMeeting), change their statuses (POST changeStatus), getTasks for an assignee (GET listMeetings), etc. It also has a POST call for modifyMeeting which lets us change the time, eventId, meeting type, etc. The main argument here is that **we'd have to force fit** our (task-assignee) data model to work with etouches' (meetingId, organiserId, List assigneeId, typeID) model.

   c. **How to process a task swap request ?**
      There are two ways to approach a swap request -
      i. **Two volunteers have agreed verbally on swapping the task, and both know their respective assigneeId (as assigned to them in etouches db)**
      Analysis : etouches provides two APIs which seems relevant at first for this task.
      PUT addAttendeeToMeeting - It only allows addition of a new assignee to the task, but we don't have any APIs that do the reverse. So we cannot add a new assignee, and remove and old one (swap).
      POST modifyMeeting - This doesn't allow a change in assignees of a meeting, rather just change in other details like location, time, etc.
      Result : Hence, the last resort that can be taken is that we change the status of the original task to something else and create a new task with the new assigneeId.
      Cons :
      1. Here both volunteers have to be present(have their attendeeId known to each other) to swap their tasks. If a volunteer is not able to find anyone in his vicinity to take up his task, he'll have to spend time finding one, or discussing with rest of the volunteers via chat.
      2. We do not have a confirmation mechanism at second' volunteer's end. In case anyone else assigns him a task without his wish, we'll have to start mechanism of reverting back those actions. This is introducing an additional overhead.

3. Etouches doesn't provide appropriate front-end that lets a moderator get notifications for changes in meetings and neither does it keep track of the same.

ii. **Second Approach is that once a volunteer finds that he'll not be able to complete a given request, he can issue a Uber-Style request, which is sent to all the volunteers with free time slot for that task.**
Analysis : For this approach, we'd need to have a list of all volunteers in a given session who are free at given time slot.
`GET listMeetings` - This API provided by etouches lets us filter on the basis of attendees, timeSlot, status, etc. However, for our purpose, we would need to filter for all the volunteers who DO NOT have tasks at a given time slot. Only way this is possible is that we get the whole list in our app, parse and load it into a local sqlite db and run a query. This could be very inefficient and expensive to run on, give the list of sessions and tasks.
Now that we got the list of applicable volunteers, we would have to add a mechanism to send a notification to all of them, which could be done via Firebase Cloud Messaging. After any of the applicable volunteers accepts the task, the we'd follow the same procedure as in **Result** of the first approach.
Cons : Since we do not have a direct access to the volunteers data, we would have to create unnecessary transformations in our android application to make it work.

d. Conclusion - As per my analysis of the etouches products, I concluded that using etouches will complicate the processing in multiple ways due to their distinct data model and limited APIs to access the data, with the only benefit being that we will not have to create front end application for moderators, which, also will be very difficult for moderators to get acquainted to, since etouches has designed the UI in their own format, catering to meeting style model.
Therefore I propose creating our own API stack over cloud server, which can very easily be created, managed and integrated into our android app. Since systers, is big organisation, I think we can very well get sanctioned an AWS or Google Cloud server to host our server application on.

2. **[Preferred] [Dropwizard/REST APIs on backend and Retrofit on Android]**

[Dropwizard] The list of tasks will be stored in the database of our choice. Since we don't require to support high bandwidth and qps from the database, a mysql db will work easily. However, it is my own opinion, and this can be discussed with mentors. Having worked a lot on Dropwizard framework for creation of APIs, I am well versed in it, and It wouldn't be a tough task to build a working REST backend. The APIs will just be executing some SQL queries for populating data in the android application, example being `getAllTasksForSessionId`, `queryStatus`, `updateStatus`, `getAllTasksAssignedToUser`, etc.
Since we'll be using Firebase for sending and managing push notifications, we can easily integrate Firebase server with our Application Server to cater to app specific requests. The implementation is discussed in the Timeline.

[Retrofit] I plan to use Retrofit HTTP Client to interact with the REST APIs on our server. Since retrofit is designed for Java and Android, the integration is seamless. Furthermore, I have worked on the similar framework while developing the Android application for our company's conference application for android.

Goto: [App Features | Technical Details | Prototype | Timeline]

Chat Room feature
(for volunteers at the same-time-same-place sessions)
I have gone through several already existing Libraries/SDK for group chatting -

1. [AppLozic SDK], 2. [pubnub-android-lolli-chat]

I modified and built the **pubnub based android chat** application on my local machine and found that it can very easily be customized to our use cases. I plan to integrate it in the hopperApp in such a way that all the volunteers are automatically added to the chat room for the current session they are managing, where they can communicate within themselves and as well as the moderators. As the session is finished, the app will automatically change the chat room to the new session, with new current volunteers are added.

We can enforce this by ensuring that we automatically generate the chat-room-name based on current session of the volunteer, and volunteers will be automatically joined in respective rooms. ***We can control what the user handle will be, and what chat rooms user can be in*** via the integration of the chat in the app. Since we do not allow the functionality of changing chat-room by the user, we can achieve the desired functionality.

Pros -

1. Servers are managed by pubnub. No server space is required at our end.
2. We require a developer key for our application to integrate *pubnub*. They have a [free version], which allows 100 daily devices and 1Mil msgs per month, which is best for our use-case.

---

6. [Optional] If you have your own project to propose, please describe it here: (Include all of the above questions if they help form your project idea.)

[Note : This wouldn't need to be a separate project apart from the hoppers App. After having laid out the whole project plan in the schedule as above, and I think I will be able to manage both - the front end, and the back end for the hoppers end to end application in the stipulated GSOC duration.]

Ans: After working on the design of the Hoppers Frontend, I am interested in the backend of the Volunteers management. If given an opportunity, I would like to work on API design and allocation part of the system, where we'll assign sessions and tasks to volunteers according to their interest and expertise, so as to ensure minimal relinquishment of tasks, while at the same time, ensuring that the 8-hour minimum contribution is adhered by all the volunteers.
According to the discussions on Slack, I came to know that the current allocations are being done without any use of technology / application. The etouches API that is being used in Conference App, ***doesn't provide much support or flexibility*** for the allocation and management of the Hoppers.
Going forward, with development of this application, I have proposed a shared data schema that is used for using Dropwizard framework for powering REST APIs that power the android app. Given that this would only require minimal maintenance on a server, developing our own APIs over the volunteer allocation database would be lightweight and provide us lot of flexibility in managing our own application. Having previous experience of Dropwizard framework for REST APIs at my work, and having them integrated in my Android Apps using Retrofit, I am confident that I will be able to complete this task too.

Systers Projects

Conference/Hopper App

If you are interested in Conference/Hopper projects, please answer question 11-12.

11. What kind of experience do you have with Android, iOS, or Windows Phone development? Do you have any design or UI experience? Describe in detail. Include any links to previous work/projects.

Ans: I worked on development of my company's android application for the technical conference, ***####***. Apart from this, I am currently developing an android application with our team of two, which is an intelligent Task Manager Application. Previously, I had participated in my college's technical festival to develop an android application for tagging geolocation using images captured from Mobile camera. As Far as UI design is concerned, I had submitted an application for developing android application for buildmlearn, but unfortunately did not get accepted. Following points contain a link to earlier application, and brief descriptions of my projects.

1. **####**
   This android application was developed for the convenience of our attendees and interested audience. I was involved in the Speakers and Schedule section of the app, where we used Retrofit HTTP Client to fetch speaker details from the our server, which was running on Dropwizard framework. These results were then cached, until an update event was received from the server. Since the conference has been completed, the app has been discontinued from the play store, but the apk is still hosted at the following address. Screencaps can be viewed at the address below.
   Link to the app : https://apkpure.com/slash-n/com.####.####

2. **butler**
   This current application is under progress and is an algorithmic task management system, where we rank the added tasks on the basis of their importance, and other features. Right now, we are using android sqlite db to store the created tasks, and processing them via our intelligent engine to get a sorted collection.
   GITHUB Link **:** https://github.com/eigencoders/butler

3. **Histreet!**
   This application was completed by me along with one of my team members during our college tech fest. In this application, we searched the image captured by the user in our image database, and estimated the current location of the user on the basis of his surrounding monuments.
   GITHUB Link: https://github.com/eigencoders/streamex-histreet

4. **Buildmlearn**
   For GSoC '15, I had created a comprehensive application for the development of the buildmlearn application, which was to be used to teach young kids to write, using advanced image processing techniques and haptic feedback. Please feel free to have a look at the wireframes and the proposal.
   PROPOSAL : github.com/shreelock/~/PPFinal.pdf
   WIREFRAMES : github.com/shreelock/~/WireFramesFinal.pdf

---

12. Have you had any experience with accessibility design? How do you develop/design applications to be inclusive for people with disabilities?

Ans: Unfortunately, I did not get a chance to do so, but I am aware of the guidelines that should be followed for our app to be accessible to everybody. I will ensure that both of our apps use an accessible color palette. Also, If required, I can integrate Google Speech Recognition to work for specific queries like "*What is my current /next task?"* where the app responds with the description of corresponding tasks, their location, and so on.

General Development and Education Questions

All questions (17-25) must be answered in this section.

17. Why do you think you are a good candidate for this project? Describe the skills you confidently bring to the project, what you hope to learn from working on this project, and your interest in the Systers mission.

Ans:

To begin with, I would have to say that I have thoroughly researched each and every aspect of this proposal for the development of Hoppers App, as it can be understood from lengthy description I have given over each feature and its implementation. I also have thoroughly researched the etouches for our application and have tried to fit in our use-case, comparing to creating our own. It is this thorough research of each and every aspect of this application, that I think will be my best contribution to this project apart from my coding skills.

As for my technical skills, I have been working for almost two years in ####, India, which has led to my exposure to various technologies and language skills, including Java, Python, SQL, scala, etc. As I have mentioned earlier, I also volunteered for development of the android application for the attendees of ####'s technical conference , **####**, where we used Dropwizard framework in the backend, and integrated Retrofit Http Client in the android app. I have been working on my own android projects in the past, but this experience taught me the peculiarities of developing a commercial android application, which is to be used by a wide variety of people.

Regarding the skill set, I have been majorly working on projects developed in Java (in the company), and have worked on dropwizard framework to create user facing APIs. I am confident that my comprehensive experience of working with wide range of technologies at ####, and my android development experience will be the magical combination that is required for the Hoppers' App Development.

Having managed projects by myself too, I am fairly confident in contemplating all possibilities of application usage and develop accordingly, which will be very useful in development of *HoppersApp*, since this would have to be developed from scratch.
As mentioned earlier, this project is different than other GSoC applications, such that it is not just another feature update, or a bug fix, but rather a totally new application, where we'll have to develop everything from scratch. This will require a deep insight in all aspects of the application, and a foresight to manage all possible scenarios. This is also require a strict end to end delivery of a fully functional system.

Systers is a wonderful organisation for women in tech, and I respect the initiatives that are taken by the community members to empower women in all aspects of life, let alone women in computing. I believe that I have a golden opportunity to put my knowledge for the betterment of the society, by getting a chance to contribute to the Systers organisation.

---

18. We have various projects in Python, Ruby, Android, iOS and Ushahidi. Describe the largest project you have completed in any of the programming languages mentioned. (Include # of members, time zones, etc.) If you haven't used any of the programming languages, describe the programming experience you have that will allow you to learn a programming language quickly and be successful on this project.

Ans:
As I have mentioned earlier in my application, I have worked on several Android applications, both in my company and as well as with my teammates. Both have been in Java/Android. If I had to talk about the biggest one so far, it will have to be the ***####*** app. It involved a team of 5 volunteers from all across the organisation and it was completed in a mere three weeks. The reason I am considering this as one of the biggest projects is due the scale to which this app was to be used. We went through multiple stages of testing to ensure data sanity and App stability.

I have also been contributing to the duckduckhack project - to improve instant answers for the duckduckgo community, where I am majorly working on python in my spare time.
***PULL REQ**:* https://github.com/duckduckgo/zeroclickinfo-fathead/pull/843

---

19. We use GitHub for our projects. Do you have experience with any version control software? Please describe the experience and list the different softwares.

Ans: Ah, Github..<3 We use github's platform and git as the version control software in my company. I have come to like **git** so much, that I do not start any of my personal project without **init-**ing it. :D
As for the experience, I have attached my github profile, and all the projects that I have mentioned above have been well managed, thanks to git!
Apart from that, I am the go to git-guy for my friends!
ME: https://github.com/shreelock
My TEAM: https://github.com/eigencoders

---

20. Describe any commitments you have over the time period of GSOC (including the community bonding period), such as classes, a summer job, vacation plans, final exams, master's thesis, etc.

Ans: I am currently working, but as I have been accepted for my Masters at Stony Brook University, I have planned to resign from the company in last week of April to spend time with my family. Hence I will not have any other commitments during the first half of the GSOC program. For the second half, since I will be moving to my college overseas, I would see a week of disturbance as I settle at my new place and start coding. This would be during the first week of august, after *Second Evaluations.* My college won't start until late August, so I won't have any other commitments until program completion.

---

21. **Education**:

- *What year are you in school?*
  - **Ans:** I am a graduate student from **####**, and am joining my Masters' program in **####**, this **####** at **####.**
- *What programming courses have you taken?*
  - *What did you like about them? What did you not like?*
  - **Ans:**
    - I am well versed in C, C++, Java**(a bit more)** and Python. I have been working at #### (India) for last two years, where I have worked in Java and scala. I learned Python and Scala on my own, and I liked them very much. Special credit to ***python***, since it is one of the few languages that do not need any additional setup to begin and finish projects. I switched from Windows to linux few years ago when I was in college, and haven't looked back since. It was due to python's extensibility, that gave me to confidence that whatever I intended to do, could be easily be done as a python script. It was at that time, when I wrote a twitter bot in python to respond a gif (using Giphy's api) depending on the reply to the twitter bot.
    - Although python is super powerful, after having worked with java and c++, I kind-of don't like that indentation is so important in python. It seems like a super pedantic compiler which would refuse to compile even if a single space is missing in the beginning of line. xD
    - Apart from this, I have taken programming courses in during our college, which included *Advanced C++ Data Structures and Algorithms, Advanced OOP with Java* as the most related to this project. Post this, my company had provided us with professional training to accelerate the learning during our induction period, where we were taught professional courses on *API design, Test driven development, Database Management*, etc.
    - Rest of the Android design learning has been completed by self motivation, with help of several Coursera and Google's courses on Android App development, and a ***lot*** of practise. :-D

- *What is your major?*
  - *Why have you chosen that?*
  - **Ans:**
    - I will be majoring in Computer Science with a specialization in Machine Learning and Computer Vision. I chose this stream since I am very much fascinated about the prospects of enabling intelligent vision like us, to the Machines.

- *Have you done group projects (programming or otherwise)?*
  - *What was your primary contribution/role in the group?*
  - *What made working in a group better than alone? What made it harder?*
  - **Ans:**
    - As much as I like the freedom of working alone, I prefer working with colleagues (even if they can just discuss with me), since they can provide me so many viewpoints and approaches to look at the problem which I wouldn't have thought by myself. I have been in multiple group projects in my workplace, as well as my personal projects.

- As far as the group projects with my team(not work) are concerned, I handle task allocation and high level project design, including discussion of various connected components. In my recent Android application, I have worked on the UI and UX design, and the algorithmic side of the application.

---

22. Do you have professional work experience in programming? Tell us about it.

Ans: I am currently working at #### ~ in the Discover team, which is responsible for intelligent selection of product results and brand banners that are displayed whenever a user logs in, or searches for anything in the App. We mainly work in Java, apart from when we have to work on machine learning models to train for user behaviour, where python is used. I have also worked with Dropwizard framework for deploying REST APIs for user facing applications, which I later consumed in the Java application using Retrofit HTTP Client.

---

23. Do you have previous open source experience? (Not including the work mentioned about Systers' projects contributions.) Tell us what you have done. (i.e. Hacktoberfest, etc.)

Ans: I have contributed to several open source projects before applying to the systers organisation. These include the **duckduckhack** repository, responsible for providing instant answers for user queries. I am currently working on a C Reference Fathead, which will be used for creating *Instant Answers* for queries like **`c printf, c qsort_f, etc.`** Before this, I had worked on **nomacs** an open source image viewer for Linux and Windows, where I added a feature to process a given image to show its Grayscale version.
PR: https://github.com/duckduckgo/zeroclickinfo-fathead/pull/843 [DuckduckGo]
PR: https://github.com/nomacs/nomacs/pull/9 ***(merged)*** [nomacs]

---

24. What would your dream job be if money was not a factor? (i.e. bookstore owner, dog walker, professional cupcake taster etc.)

Ans: If money wasn't the issue, I would want to be a designer. I pay a lot of attention to small details that are put in the things that we use day to day. For example, we haven't yet achieved much success in developing a speed bump/barrier to enforce a one way driving rule for our roads. These are some of the things that spark my imagination and my will to have intuitive and brilliant designs.

---

25. Schedule a 15 min interview with Maybellin Burgos. Fill out this whenisgood and May will reach out with a Google Calendar invite.
