GSoc '18 Project Proposal

General Improvements and Bug Fixes for SUSI.AI android app

PROJECT SUMMARY

The main aim of my project is to solve various major issues in the SUSI android app and at the same time also extend the features of its app in various ways which can help improve the user experience and ultimately the productivity of the app. I also intend to take up some ideas from the ideas page of fossasia and inculcate them in my project which can provide the best features for the user.

BASIC INFORMATION

Personal Information

- **Name:** Aniket Agarwal
- **Email:** agarwalaniket1999@gmail.com
- **Telephone:** 9650788548
- **Time Zone:** India (UTC+5.30)
- **Github UserName:** aniket-agarwal1999
- **University:** Indian Institute of Technology, Roorkee
- **Major:** Applied Mathematics
- **Current Year:** 1st year (2022 expected graduation)
- **Degree:** Integrated Master of Science (5 years)

Coding Skills

- Fluent in Java, Kotlin, XML needed for Android app Development
- Also fluent in C++ and beginner level experience in Python
- Basic knowledge of HTML, CSS, Javascript and Shell Scripting
- Strong experience in Git

Development Environment:

- Ubuntu 16.04 LTS or Windows 10 (I work on former most of the time)
- Fully customized Android Studio for efficient use.
- Microsoft VS code as primary editor

PRE-GSOC INVOLVEMENTS

Contribution to susi_android

By thoroughly going through the codebase for Susi android app, I found some bugs and also some additional things that needed to be integrated in the app on urgent basis. To make these issues go away I have submitted various PRs. Also I actively tried to review the PRs submitted by my fellow co-developers and also tried my level best to help them out.
Some of my contributions are:

Pull Requests:

- #1183 **(Open):** Fixes the display of search option in skill details, skill listing fragment and also a minor UI tweak. Fixes #1177
- #1197 **(Open):** Changes the UI of Skill Listing, Skill details Activity. Fixes #1196
- #1202 **(Open):** Addition of a delete chat history option in the app. Fixes #1201

Issues:

- #1176 : Display of 'Today' rather than the date for all the conversations held in the present date.
- #1200 : Addition of try the skill button with every skill in the skill listing page.

In addition to these, I will keep on contributing to the repository before the official time for the project starts.

DETAILED DESCRIPTION

Adding an In-App browser:

An In-App browser can surely help the user get more engaged in the app as rather than the link opening in some other browser, we will give the users a feature of the link being opening in the same app.
So whenever the user clicks on a link in the app, he will be directed to the in-app browser rather than the default in the user's phone. This can definitely help in increasing the productivity of the app as the user would then only be engaged in our app rather than going out of it.

Implementation:

For this purpose, we can use **WebView,** which is already present by default in the android studio.
The In-app browser would have some of the essential features like:

- Refresh option
- Share the webpage option
- Go back and forward option: Can be implemented by using **canGoBack()** and **canGoForward()** methods in webView.
- Zoom In and Zoom Out option to be given to the user: Can be implemented by using **zoomIn()** and **zoomOut()** methods in webView.

- A close button at the top of the menu bar so as to close the browser.

Now so as to integrate this in-app browser in our app we can just change the code snippet shown below from opening an Intent to rather just opening the **linkurl** in our web browser by calling the java class for the in-app browser

```
        holder.previewLayout.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (linkurl != null) {
                    Uri webpage = Uri.parse(linkurl);

                    Intent intent = new Intent(Intent.ACTION_VIEW, webpage);
                    if (intent.resolveActivity(context.getPackageManager()) != null) {
                        context.startActivity(intent);
                    }
                }
            }
        });
    }
}
```

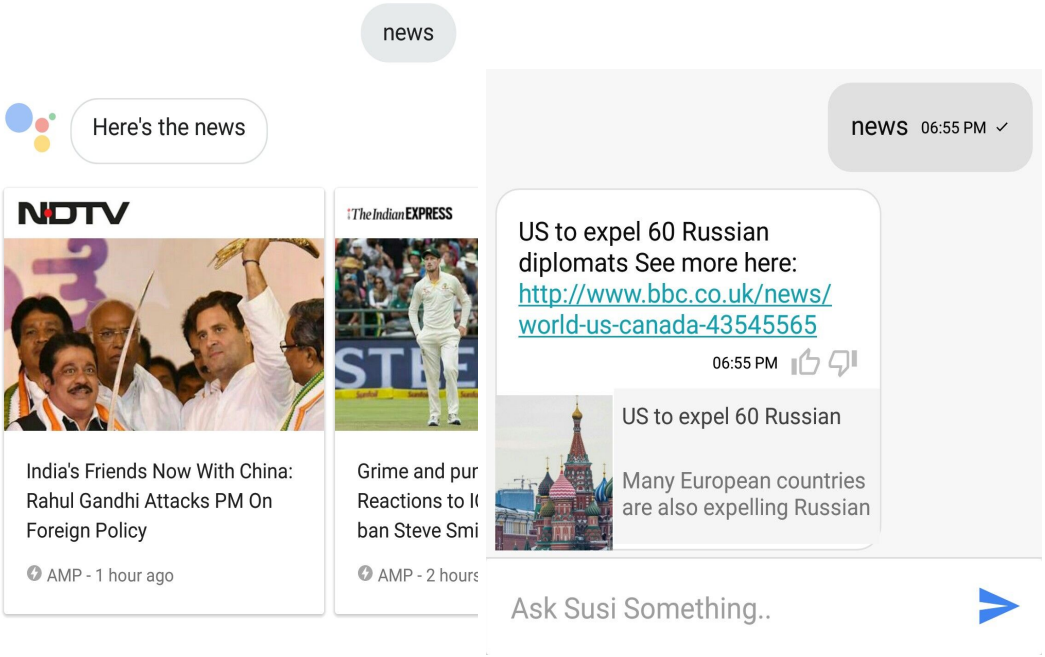
This is there in the recycleradapters directory which is there inside the chat directory

I will be making the webView taking inspiration from webView of various organizations like Facebook, Google, Twitter, etc, and wish to incorporate some of their best features in my app. So the browser would look a bit of this sort:

This is taken as a reference from the in-app browser of Microsoft Cortana app

Adding a RecyclerView for a horizontal display of news and images related queries in the app

Presently if the user drops query such as news and images, rather than displaying a range of news or images, Susi only displays one news/image at a time. This is quite in contrast with the Google Assistant's response to such query.



A comparison between the presentation of news in both the apps

Implementation:

To implement this change I would have to make a custom recyclerView(s) for image and news display. Like the **ChatFeedRecyclerAdapter** in the app, I would be making an adapter for the sake of this particular response. Also presently for images the response only contains a link of the image and hence I would like to improve that by also showing the image in the view itself and also making a recyclerView for its horizontal scrolling.

Adding a Floating widget for the app:

We can easily implement a floating widget for our android app, just like Facebook messenger chat head, and hence give the user a chance to just open and chat with Susi when he/she is on any screen. This can certainly improve the user experience as they will be able to search for a particular keyword or anything without specifically opening the app.

This is the basic implementation of how the Chat Head would look like(Many changes would be done)

Implementation:

It can be implemented by creating a custom FloatingWidgetService class which can handle all the operations related to the Floating widget. One of the most essential features of this class would be to override the **OnTouchListener().** Whenever the user touches the chat head, we will record the initial x and y coordinates, and when the user moves the finger, the application will calculate the new X and Y coordinates and move the chat head. So we have to handle the various operations:

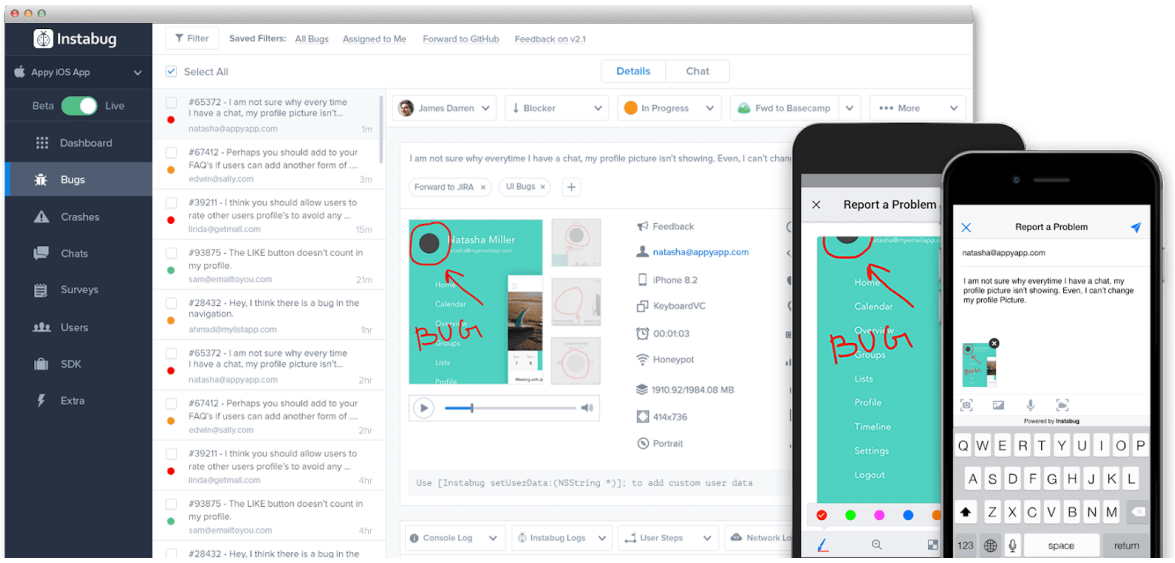
- Triggering long click if user holds or press chat head for a long time.
- If long click event occurs then displaying RemoveView(A small cross at the bottom of the screen) visibility.
- Trigger on click event of the chat head.
- Stopping Floating widget service if user drags and drop the Floating view into Remove View.

Also to implement a chat Head we will also need to set the android permissions so as to draw the let the widget be visible over other apps.
Also the user can be given an option in the Settings page of the app, so as start or stop the widget service.

Reporting an issue by the user

Reporting of issues by users of our app can certainly lead to a better user experience. Currently the only way for users to report a issue related to app is to make one on the project's github repository, which certainly can be quite intimidating for a fellow user. So to ease out this process of issue reporting we can directly use the services of **Instabug.** By integrating this in our app we can just directly give the user an option to report a bug related to the app in just some simple steps.
Some of the advantages of this can be:

- **Fix Bugs faster**
  Instabug captures user steps, network logs, view hierarchy and all device details to help you know exactly why a bug happened, so you can spend your time fixing rather than debugging.
- **Get Product Feedback**
  Allow your users and beta testers to send their feedback and report bugs, directly from within your app, by attaching screenshots, voice notes or even screen recording to better express themselves.



Implementation:

According to the documentation of **Instabug,** we can integrate its services quiet easily in our app:

(This we have to integrate in the build.gradle in our app)

```
<dependency>
    <groupId>com.instabug.library</groupId>
    <artifactId>instabug</artifactId>
    <version>4+</version>
</dependency>
```

(This we have to write in the onCreate() method of our activity)

```
new Instabug.Builder(this.getApplication(), "05da030a5617730fffe4e31846c0f038")
    .setInvocationEvent(InstabugInvocationEvent.SHAKE)
    .build();
```

By doing this the user can just submit the bug by mere shaking of the phone. With the help of this issue tracking system, the user will be given various options:

- **Screenshot Annotation:** Users can attach screenshot with their issue.
- **Screen Recording:** Users are also given the option to record the screen and send them with their issue.
- **User Profiles:** The user profile will display an array of details about users and testers. You will be able to know the app version they are on, the custom fields data and the timeline of their application usage.
- **Integrations:** We can easily integrate the Bug Reporter with various applications like Github, Slack, Gitlab and many others.

- **User Steps:** We can easily know everything that user did before ending up with the specific bug.
- **Network Logging:** Whenever the bug occurs due to server issues, you will know exactly what happened with the network request and responses till the moment the bug was reported to be able to identify what needs to be fixed.
- **View Hierarchy:** It can make the process of finding weird UI bugs in the app.

So its capability of being integrated to Github gives the extra advantage to all the developers out there who are testing the app in beta stage, by reporting a bug with extra ease and with a better description.

Adding Login option using Facebook, Google, Twitter, etc in the registration part of the app:

We can integrate Facebook, Google, Twitter, etc. logIn methods in our app. So in addition to the usual Register and Sign In method we can create buttons so as to give user the feature to Sign In using their pre-existing accounts on various popular sites.

Implementation:

To do this we can just use the already existing APIs.

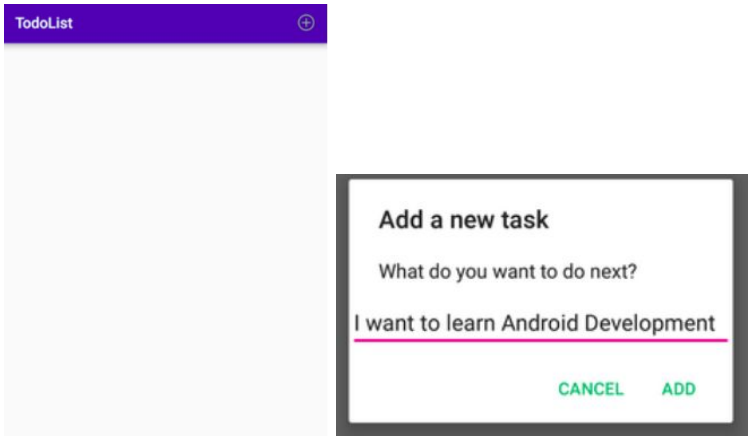
- To integrate **Google Sign-In** for the app we can use **GoogleSignInClient** object in our app. So now I will also call functions that can also get an already once signed in user. Apart from this we also have to add various dependances in our app so as to be able to start off with use this feature. Also the Google Sign-In gives an option to fetch the email and Id for a new user so that we can also save the new user's credentials on the server side.
- For Facebook, Twitter and many other app's log In also we can follow similar steps and follow their documentation.

Creating a To-Do list in the app:

Now one of the basic functions of a Virtual Assistant is to keep a track of all the important things in the app. However, Susi still fails in this front as it does not provide the user a way to store some of his/her important things.
So I would like to implement this feature in the app by making a To-Do list in the app by taking the help of Realm Database to store the list locally in the device itself.

Implementation:

Now to implement this, I would be creating an a FAB just like that of Susi skills, which upon clicking will open the ToDo list activity where user can see and edit the List by removing or adding on new tasks. To implement this we can use Realm Databases.



Now so as to make the best use of this To-Do list we can even create a widget for our app which will basically be showing the To-Do list stored in our app, quite like Google Keep.

Widget Creation:

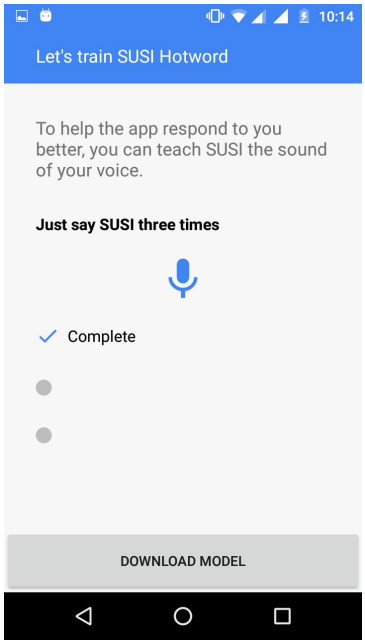
So as to increase the functionality and productivity of Susi android app, we can create a widget so as to implement the widget creation for our app. The function of this widget is to basically show all the items of To-Do list in our app. Also by clicking on a **(+)** button on the top of this widget would straight away get the user to the create new task fragment of To-Do list activity of our app. And clicking on any of the tasks would get the user to the task list.
To implement the same, I will be using the in-built functions in the android studio which can enable widgets for the app. The aim of this widget is to look quite like the widget of Google Keep.

Customizing the hotword for every user:

Hotword Detection is undoubtedly one of the best and unique feature present in the Susi app. However still due to the absence of enough training for our model, it leads to lagging for different users due to their different accents. To solve this we can rather than configuring a pre-existing .pmdl file, we can rather create a .pmdl file for every user.

Implementation:

- We can use the same kind of implementation and UI used by Google Assistant when configuring our voice for its hotword detection.
- So to do this, Snowboy provides an API to train and get a pmdl file. We can prompt the user to speak the hotword **"Hey Susi"** three times. These three recordings will be sent to this http://docs.kitt.ai/snowboy/#api-v1-train API as a post parameter. The API will return a .pmdl file which then will be used for hotword detection.
- Once we have a universal model trained by 500 person we can update the app and use universal model which will work for everyone and there won't be a need for every new user to train the model.



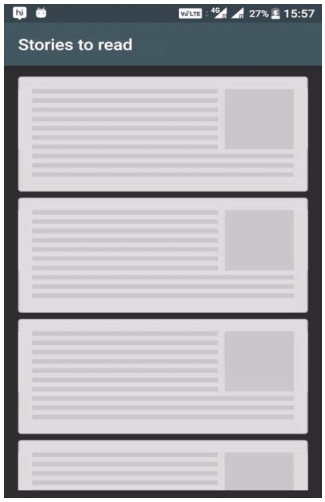
A sample of how the UI will look like

Reference from a pre-existing PR([link]) still in works.

Major UI changes:

The UI of the Susi android app is certainly quite good but compared to its some of the biggies like Google Assistant, Microsoft Cortana, it stills lags by a huge margin. So I would like to improve the UI of the app so as to improve the user experience and also make them more engaged in the app.

1. **Shimmer RecyclerView:**
   Now this is the component that can be found in most of the android apps that gives an animation effect to the loading RecyclerView cards. This can most certainly be used in the skill listing page of the app so as to animate a particular skill card on the page.



Now to achieve the same there is a library related to this particular thing on Github (link). This can be integrated in the app and can be used in the skills page.

2. **Changing the UI of the whole app to follow the Material UI standards and extending the information displayed in the About Us page.**
   Right now, the thing that the UI of the app lags in is the absence of the Material Design standards at various places. So I would like to change the present UI of the app.
   For this purpose, I would like to take inspiration from Google Assistant UI and hence make some major changes in the UI of the app.
   In addition to this I would also like to extend the information displayed by the About section of the app to match it with the About page as displayed by web app of SUSI.AI

Major Bug Fixes:

I will also be solving some of the bugs in the app that require Urgent attention so as to make the app's interface user-friendly.

1. **Skill info and examples are not loading** (Issue: #1008)
   An error occurs whenever the user opens the Skill Listing page displaying that the Skills cannot be fetched. When the Logcat was checked by me, it displays this to be a result of **NullPointerException.**
   To solve this, I can use debugger and can find the exact position where the exception is occuring and hence try to fix it.

2. **Improving the Code quality**
   One of the major problem faced by the new contributors to the organization is getting lost in the large codebase of the app. So as to solve the same I intend to add comments in every file of the codebase so as to ease out the process of understanding the code for the beginners.

3. **Fixing all the typos, extending the documentation**
   Although the documentation of the project is quite good enough, but the documentation still requires an extension considering the fact that various new features like Hotword Detection, etc are also added in the app.

4. **Adding the Search option in the chats activity:**
   Now to let users search for a particular message in the Chat History I intend to add an option which can help the user search a particular keyword or a phrase in the Chats Activity of the app.
   To implement the same I will be using a **Android's search view widget.** For the UI we will be using button on top of the chat activity screen so as to initiate the widget and then the searched keyword will be highlighted on the screen.

5. **Improving the Delete chat option:**
   Now I have already implemented a delete chat option in one of my pull requests(#1202**),** but this is only able to delete the chats from the local database and hence gets restored once the user logs out and once again logs in. So I would like to further enhance my PR by even deleting the chats from the server side.

Project Timeline

My main focus will be to enhance the various new features specified by me. So approximately **70%** of my time will be utilized in implementing these features. **20%** of the time will be utilized in implementing the UI changes and the remaining on the Bug fixes. Except for this, I will also try to solve various issues reported by other developers and users throughout the project.

Duration	Milestone
April 23- May 13 (Will be busy with Semester exams from April 23- May 4)	● Will do an in-depth analysis of the codebase of the app so as to get familiar with it and also find any more issues related to it. ● Interacting with the mentors and getting their views related to the UI change and also the things they want to be included in the new UI. ● Also solve any minor issues reported on the github repo of the app.
May 14- May 20	● Solving the issue of the Skill Listing activity showing error(Error: The skills cannot be fetched). ● Improve the code quality by adding comments everywhere.
May 21- June 3	● Complete any pending work from previous week. ● Implementing the “Report an issue” option in the app. ● Testing the Reporting issue feature rigorously and even integrating it with github.
June 4- June 10	● Adding login option using Facebook, Google, Twitter, etc. in the registration part of the app.
Phase 1 evaluation (June 11 - June 15) **Major deliverables**:	

<table>
<tr><td colspan="2">1. Add the “Report an issue” option in the app
2. Adding login option using Google, Facebook, Twitter, etc.</td></tr>
<tr><td>June 15 - June 28</td><td>● Creating a To-Do list in the app(including a widget for the app).
● Adding the search option in the chat activity.
● Improving the Delete chat option.</td></tr>
<tr><td>June 29 - July 5</td><td>● Customizing the hotword “Hey Susi” for every user.
● Completing any previous task.</td></tr>
<tr><td>July 6- July 12</td><td>● Make the in-app browser for the app.
● Adding a RecyclerView for the horizontal display of news and images in the related queries in the app.</td></tr>
<tr><td colspan="2">Phase 2 evaluation(July 9 - July 13)

Major Deliverables:
1. Creation of a To-Do list in the app.
2. Customizing the hotword “Hey Susi” for every user.
3. Making the in-app browser for the app.</td></tr>
<tr><td>July 14- July 27</td><td>● Adding the floating widget for the app.
● Implementing the shimmer RecyclerView in the Skill listing page.
● Changing the UI of the whole app so as to follow Material UI standards</td></tr>
<tr><td>July 28 - August 5</td><td>● Fix any minor bugs or issues.
● Extending the documentation of the app by incorporating all the changes made by me during the project. Also fixing all the typos in the project.
● Completing any previous task.</td></tr>
<tr><td>August 6 - August 14</td><td>● Final testing of the app.
● Final report submission on the work done.
● Submission of the project</td></tr>
<tr><td colspan="2">Final Evaluation(August 14 - August 21)

Major Deliverables:</td></tr>
</table>

1. Adding the floating widget of the app.
2. Changing the UI of the app.
3. Final report of the whole project.

Apart from all these I would be very much interested in implementing any ideas suggested to me by the mentors in addition to these features.

ABOUT ME

I am an 18 year old, first year student currently enrolled in Applied Mathematics (V year course) at IIT Roorkee. I developed a passion for programming in my school days, where I was fortunate enough to learn basic C++. However my learning curve achieved great heights only after coming to this institute where I finally understood the practical application of my knowledge.

Being a newbie to the open source culture, I am quite amazed by the beauty and vastness of the same. I am constantly in a search to learn new things in the field of programming and open source has been a major help to me in this area.

MOTIVATION

My motivation for GSoC this year is getting myself more familiar with the Open source organizations. GSoC is a great program for introducing organizations with prospective contributors and when I saw this project, I thought this is something that I can do. I believe this project can definitely help me to extend my boundaries, as this would be the first time I would be working on a big project with a real world impact. This prospect is very exciting to me.

I chose this particular organization because it is very well-aligned with my interests and also correlated to what I have worked in the past. Other than this, this project provides a good opportunity to apply my learning on a practical scale. This is my field of interest and therefore the natural inclination to this project.

AVAILABILITY

So as to fully concentrate on this project only, I have only applied for this organization as a part of GSoC 2018. Also I don't have any specific plans or internships planned out for this summer. So I can easily devote around 45-50 hours every week very easily during the program as I like to code even in my leisure time. After the opening of my institute on July 17, I will be a bit busy with my classes, but I would complete most of my work before that time and also can easily devote 35-40 hours in that period.

PROJECT OUTREACH

Being a part of such a vast community is a great opportunity in itself and I would love to collaborate with others throughout my project timeline and even after that, as this is the true essence of Open Source culture. I would constantly contribute to the community by helping various other Open Source enthusiasts who wish to contribute by helping them out with their problems and at the same time encouraging them for participation. Also I would be submitting various reports on the work done by me on a regular basis. Under the constant guidance of my mentors I would like to make the best out of my summers and do my part towards making this organization even better. Also I would be very much pleased to keep on contributing to this organization even after the program by being a mentor for Google Code In and various other events organized by the organization so as to help various out there in starting there open source journey.

References

1. https://developers.google.com/identity/sign-in/android/sign-in
2. https://developers.facebook.com/docs/facebook-login/android#quickstarts-header
3. http://www.androhub.com/android-floating-widget-like-facebook-messenger-chat-head/
4. https://developer.android.com
5. https://www.sitepoint.com/starting-android-development-creating-todo-app/
6. https://docs.instabug.com/