

Google Summer of Code 2020

Proposal For SugarLabs

Debian Advocacy for Sugar

Basic Details

Name: Shaan Subbaiah B C

Email: shaansubbaiah.cs18@bmsce.ac.in

Github: <https://github.com/shaansubbaiah>

IRC: (freenode) shaan_subbaiah

Phone: +91 9980213650

First Language: English

Timezone: Bangalore, Karnataka, India; +05:30 UTC

Project Details

Synopsis

The project revolves around the packaging, testing and debugging of Debian (Experimental, Release, Stable versions) with Sugar and its dependencies, checking the functioning of the core features of the application, online collaboration, save and resume features.

Impact on Sugar Labs

Sugar Labs empowers and educates children, with a focus on collaboration, reflection and discovery, through their Sugar software platform.

The successful completion of the project could result in the Sugar packages being available in the future Debian releases. Some of the most popular distributions of Linux like Ubuntu, Mint and MX Linux are based on Debian. This would help in making Sugar more accessible to millions of people.

Technologies to be used

The testing and debugging of the Sugar would require knowledge of **Python**, **GTK+ 3**. I will be testing Sugar, the toolkit, activities, collaboration between network instances. I would also be using:

- **Virtualization software** like VirtualBox and KVM to test collaboration between multiple network instances.
- **Macro/ automation software** like xdotool and AutoHotkey to check for user interface issues upon input.
- **Core Packaging tools** provided by Debian such as dpkg-dev, debconf, fakeroot.

- **Package lint tools** such as Lintian, piuparts, debdiff to help in finding common problems and policy violations in the packages.
 - **Other tools provided by Debian** to help build, upload and automate the package building process such as git-buildpackage, dupload, devscripts, etc
-

Packaging

Packaging Sugar for Debian

Package files used/ built

- **Upstream source archive file:** The Sugar package source tar.gz / tgz file from which the actual Debian package will be made.
- **Source package:** Addition of appropriate Debian files such as to make it compatible with the packaging tools.
- **Binary package:** The final Debian binary package .deb that will be installed.

Testing of built package

Run a chroot environment - changes apparent root directory for running applications, prevents breaking the developer system, makes it easier to find missing dependencies. Run tests:

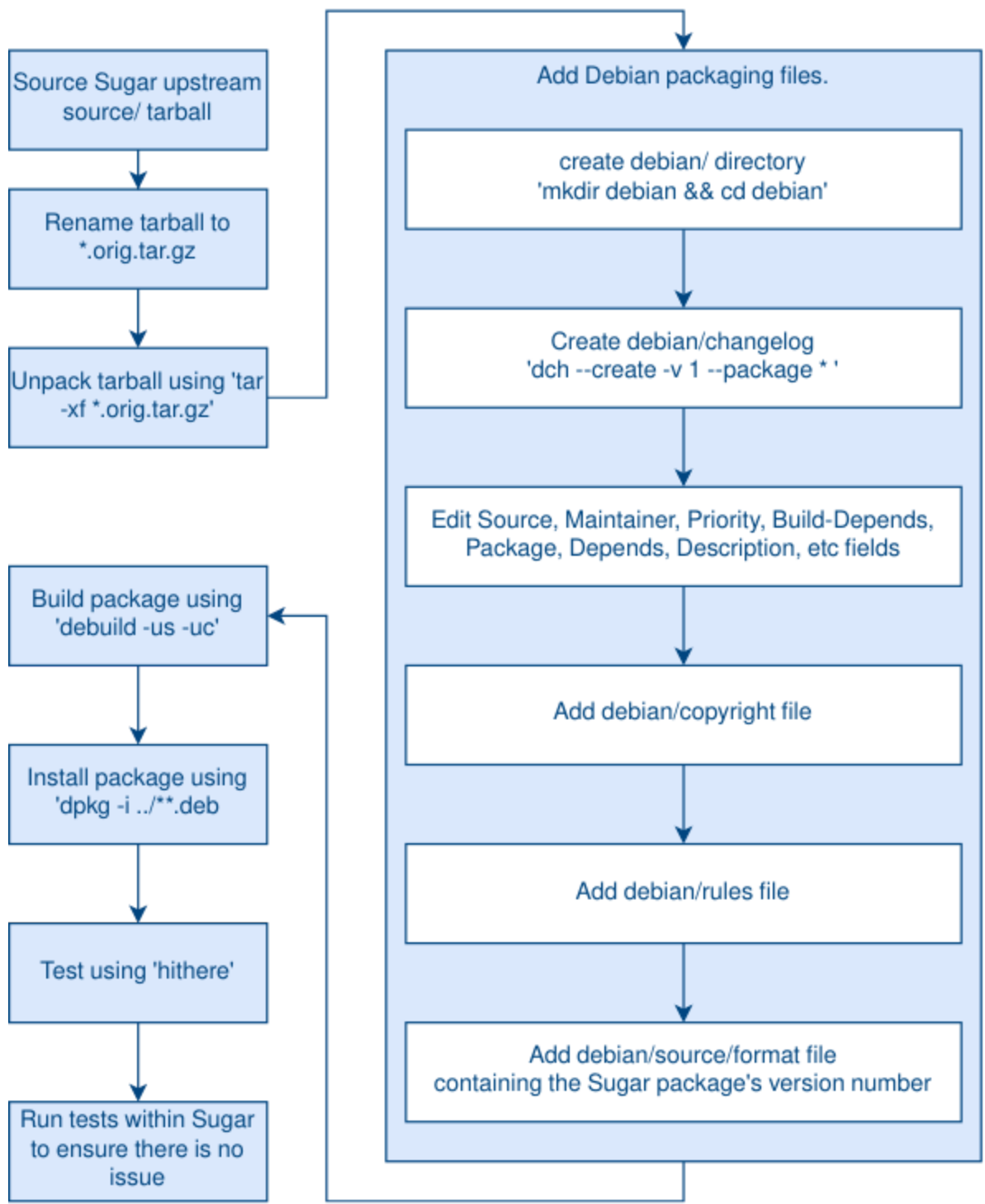
- **Using Lintian**

```
lintian package-version.changes
```

- **Using Piuparts**

`piuparts binpackage-version.deb`

- Automated checks using check-all-the-things Debian package



Debian packaging workflow

Testing

Testing of Sugar packages for Debian

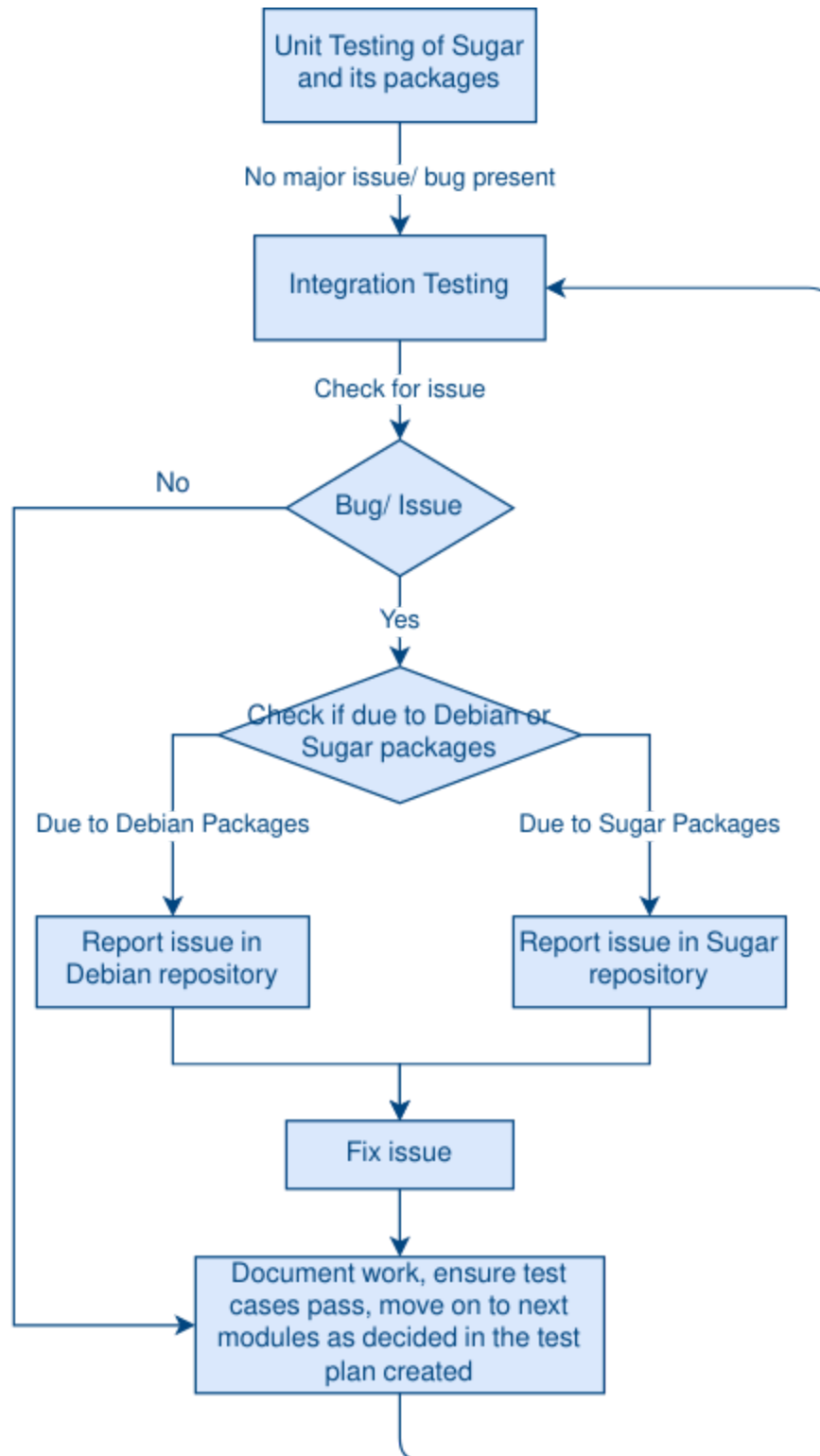
Setup of testing environment

- Setup appropriate environment for testing Sugar and its packages on the various releases of Debian (Debian latest release (Buster), testing release Debian 11 (Bullseye), and experimental release) using a Virtual Machine/ other means.

Features to be tested

- Application should start and shut down normally
- Major features of each Activity
- Collaboration to be testing if supported in the Activity
- Major issues breaking the user interface
- Save and resume features of the Journal
- Working of clipboard, date, colors, etc as outlined under [Organized Test Cases in the Semantic MediaWiki](#)

Entry Criteria	Exit Criteria
<ul style="list-style-type: none">• Unit testing Sugar packages• Created test plan, test scenario to be documented• Ensure test environment is set up	<ul style="list-style-type: none">• Successful testing of Sugar packages• Ensure high priority bugs/issues are fixed• Documented executed test cases



Project test workflow

Project Timeline

This is a rough timeline, as the project revolves around testing, it will likely change after discussing with the mentors, creating test plans and unexpected issues.

Community Bonding Period (May 4th - June 1st)

During this time I will get to know the community better, converse with the members, familiarize myself better with the code base and set up a workflow to optimize my productivity for the coming months.

Coding Period (June 1st - August 24th)

Duration	Milestone
May 21st - June 4th	<ul style="list-style-type: none">• College Semester End Examinations
June 5th - June 14th (weeks 1, 2)	<ul style="list-style-type: none">• Setup test environment• Go through Debian packaging documentation• Brush up on Python, go through PyGObject• Discuss with mentors and decide on best testing approach• Document work done
June 15th - June 29th (weeks 3, 4)	<ul style="list-style-type: none">• Creation of Debian packages for Sugar• Discuss metadata required for packages with mentors• Fix issues encountered during packaging

	<ul style="list-style-type: none"> • Test packaging as outlined in the flowchart created above • Document work done
1st Evaluation (June 29th - July 3rd)	
<p>July 3rd - July 16th (weeks 5, 6)</p>	<ul style="list-style-type: none"> • Create test plans • Test if Sugar installs, starts and shuts down without errors • Test major features of fructose activities • Report and fix issues found • Document work done
<p>July 17th - July 27th (week 7, 8)</p>	<ul style="list-style-type: none"> • Fix pending issues • Test collaboration features • Test save, resume features of the Journal • Run required tests from the Wiki's Organized Test Cases • Document work done
2nd Evaluation (July 27th - July 31st)	
<p>August 1st - August 14th (weeks 9, 10)</p>	<ul style="list-style-type: none"> • Fix pending issues • Regression testing • Document work done
<p>August 15th - August 24th (weeks 11, 12)</p>	<ul style="list-style-type: none"> • Utilize the last 2 weeks to ensure all major bugs are fixed • Complete documentation
Final Evaluation (August 24th - August 31st)	

Availability

I will be able to spend at least 7-8 hours a day, 6 days a week working on the project. I have my Semester End Examinations from which have been marked in the timeline, during which I would be able to work 1-2 hours a day, 7 days a week. I do not have any other projects/ Internships lined out for the summer so I will be able to dedicate my time completely towards this project.

Due to the current situation with COVID-19, the dates for my examinations mentioned earlier may change. I will be notifying my Mentors as soon as I am informed of the same.

Communication

I would prefer reporting my progress via the Sugar-devel Mailing List and/or using a blogging platform such as Medium/ Github Pages. Open to use any other medium if required.

Post GSoC

I plan to contribute towards other Ideas that have not been taken up during GSoC, write documentation and maintain my work done during GSoC.

About Me

I am in my sophomore year, pursuing my Bachelors of Engineering in Computer Science at BMS College of Engineering, Bangalore. I have a deep interest in Computers and open source software. I have been using Manjaro as my daily driver for the last 3 years, I enjoy the flexibility and customizability open source software provides.

Also passionate in design, I spend my free time theming the Plasma desktop and tinkering with custom ROMs on Android devices. I am part of the design team that organizes my college's National level technology fest 'Phase Shift' and cultural fest 'Utsav', some of my digital art can be found on [Instagram](#).

Skills

Languages: C/C++, Java, Python

Frontend: HTML, CSS, Javascript, Node.js

Backend: Building REST APIs using MySQL and Sequelize

Regarding past Open Source experience, I took part in Google Code In 2018 and a few design competitions held by KDE. This is the first time I will be contributing through GSoC.

Work Experience/ Projects

Software Developer Intern at KarmaLife (AI Foundry)

Worked directly with the Data Science team in developing a dynamic dashboard to provide a user-friendly, simplified view of data and help the company make smarter, data-driven decisions.

Dashboard built using Python3 - Plotly Dash, Flask, Dash Bootstrap Components. Fetched data from mongoDB

Open Source Projects:

Nimbu - Music Bot for Discord ([Github](#))

Discord bot that allows users to stream audio from Youtube. Supports various playback commands and features generally available in paid bots such as client side volume adjustment, a dynamic queueing system and saving audio offline. Features a command handler, making addition and removal of features a breeze.

Uses Node.js, Discord.js, fluent-ffmpeg for transcoding audio, ytdl-core for sourcing audio and yt-search for audio metadata.

Book XChange - Platform to sell or donate books ([Github](#))

Intra-college online platform for selling or donating previous semester's books, notes, drafters, lab uniforms, etc. Backend is still in development.

Built using Node.js, Express, Material.io

Other smaller projects available on [Github](#)

Previous contributions towards Sugar Labs:

- Report issue where Write Activity window could be dragged in Ubuntu 20.04 packages <https://github.com/sugarlabs/write-activity/issues/38> **[Fixed]**
 - Fix installer script for Sugar v116 on Arch Linux by Srevin Saju <https://github.com/srevinsaju/sugar-arch/pull/1> **[Merged]**
 - Fix issue due to Python 3 Type Error <https://github.com/sugarlabs/sugar/issues/909> **[In Progress]**
 - Create documentation to help new users to 'login to Sugar' <https://github.com/shaansubbaiah/tutorial-sugar-log-in> **[Merged]**
-