Build Sugar App Store Website For Python 3 Activities

Personal Details

Personal details removed as sharing proposal on mailing list. Rest of the proposal is same as submitted for GSoC 2020.

Synopsis

There is currently an app store website for python 2 activities written in python, php and using MySQL database but there is no functional app store for python 3 activities as development on app store written in php stalled. App store for python 3 activities will become even more crucial as python 2 based activities would be phased out since python 2 official support has ended. Students and Developers can benefit from app store which serves their need and bring app store to Sugar Desktop Environment/Learning Platform similar to app store on other desktop environment and operating systems. The project I purpose delivers working app store within first month with basic skeleton and add additional features incrementally. App store will be build by gathering requirements from the developers in community and understanding the need of students. Once we arrive at agreed requirements specification and hi-fi prototype of app store, development work will be carried out.

Project Benefits

The website will serve dual purpose of allowing users to browse and add activities to their system and developers to uploaded and update activities. It is important to have activities in an organized, browsable and searchable app store from ease of use and security perspective. It is safer and easier for students to browse and install activities from one platform rather than searching on internet such as on GitHub and adding from their. This greatly reduces the risk of student accidentally executing malicious script from internet.

Furthermore, desktop app can be created for Sugar desktop environment with marginal effort. Therefore, bringing native app store to Sugar Learning Platform, which will be helpful to students to easily and safely browse and add new activities to their system. The difference between a website and desktop app would be that, website caters to both developers and students while desktop app will cater to users/students only for adding activities, and additionally for removing activities.

For me, this is a great opportunity to join FLOSS community with first time *code* contribution to existing organization. To be clear, I am publishing all my software from start under free software license and I have made significant contributions to OpenStteetMap, however not by coding but by mapping. In proposed project, I work on a new project from scratch with minimal interaction with existing codebase. This will make it easy for me to participate in Sugar Labs community with valuable contribution as I become more familiar with existing codebase and learn to work in collaborative coding environment. I chose to participate in Sugar Labs community because of your work's value in education of children. Once project completed, I will be very happy with my contribution to the childhood education.

Work Breakdown Structure and Deliverables

Work is broken down estimating 20hr/week workload while my classes are in session and 40hr/week during mid-term break i.e. 26 June to 27 July (Week 5-8). Very conservative estimates are made so as to not over promise and under deliver. Also, since documentation and, testing for codebase coverage and features will be done from the beginning, process would be slower but will produce better quality and reliable software.

April 1-14: Requirements Gathering

I will communicate with the mentors and wider developer community of SugarLabs to understand the objectives we wish to achieve and all the tasks that are agreed to be completed. Communication can be done via mailing list, IRC or gitter. For faster communication, I request some discussion to be organized over IRC or gitter.

Deliverable: Requirements Specification Document

Milestone: Clear understanding of requirements and what needs to be done

April 15-28: App Store Workflow Prototype Iterations

Based on requirements and understanding needs of students and activities developers, high-fidelity workflow prototypes will be built. Community and mentors can browse through this prototype and suggest changes and improvements. There are various benefits of first building non-functioning (but simulating) prototype and making changes to it such as ensuring everyone is on the same page on functionalities and appearance of app store, will serve as a guidance for development, as it will be exactly known what the final app store will do and look. Therefore, development would not go astray and can measure progress against prototype. Lastly, making changes to prototype will be much faster easier and than making changes to code at later stage.

Deliverable: Final high-fidelity workflow prototype of app store

Milestone: Prototype of app store built

April 29-May 5: Software Design: Technologies, Frameworks, Architecture, High-level Test Cases

Software system design will be built, identifying technologies (such as MySQL & Nginx), frameworks (such as Flask or Express.JS for backend), libraries (such as template engines) etc. that can be used, and will be selected upon discussing with mentor. Deciding upon architecture of software such as microarchitecture, modular or model-view controller. High level and integration test cases will be written.

Using MySQL DBMS would be a perfect balance between scalability, ease of implementation. We may use flask (python) or express.js (JavaScript) framework for backend. Both are simple, lightweight but powerful and I have used both of them. From project idea, it appears as python would be language of choice for backend.

May 6-19: Basic App Store Functionality Built

During this time, some sample activity bundle will be uploaded manually to test but does not have system for become automatic app store for all available python 3 activities in Sugar ecosystem

- Redirect Sugar 0.112 running on Fedora 18 or earlier version users to python2 activities app store based on their user agent

- Skelton backend and Frontend with sample activity bundles displaying their details, listing sample activity bundles by various metrics such as category, download option of activity bundles, https support, reverse proxy setup etc.
- Software detailed layout blueprint, interaction between database, backend, activity bundles stored on drive, ssh request server etc.
- Write unit test and test cases and run, measure satisfactory test coverage

Deliverable: Design Document illustrating software architecture and detailed organization

May 20-31: User Interface

Work on UI: write CSS, Write and test for responsiveness metrics, find/modify/design Icons etc. Write unit test and test cases and run including previous tests, measure satisfactory test coverage of CSS.

Deliverable: Working app store from user's perspective

Week 1 & 2: Process Uploaded Activities And Browsing & Search Functionality

- Code to extract activity data such as name, summary from activity/activity.info and add to database and store source code (if needed) & activity bundle in directory
- Implement listing and search functionality using database capabilities and feeding to page template

2nd and 3rd week of June, I would have had final semester exams. Though as per recent update from university on 31st March, due to COVID;-19 pandemic, these exams will not be conducted and alternative assessment criteria will be decided. Because of this, I cannot tell in advance if it will reduce my ability to work on this project during Week 2 and 3. I will update mentor well in advance, as I get more information.

Week 3: Upload Activities Via SSH

Implement functionality to add or update activities on app store via SSH. Should only the ID which created an activity be able to modify it? If so, such functionality will be implemented. ID used will be created manually at backend i.e. their entries will be added to database directly. Should there be a registration system for developers and approval system for activities? Than it will be implemented in extra time available since Week 9.

Deliverable: A Fully functional standalone app store

Week 4: Microformat Software Upgrade Support

Add support for Sugar's microformat software upgrade feature, so that python 3 activties can be upgraded from this app store

Work before it to be completed before Evaluation 1

Week 5: Automatic Update Of Activities On App Store

Implement functionality for automatically detecting new releases on GitHub and other common hosting platforms of activities, creating bundle from source code, extract and display release notes

Milestone: Automation of adding new release of activities to app store

Week 6 & 7:

- Implement feature to detect if an activity is installed and display so in web app.
- Changes to Sugar's software upgrade feature to point to this new web app store.

Milestone: Integration with Sugar Learning Platform

Week 8: UI Revision

Refining UI based on feedback received so far, styling beyond basic user interface, implement miscellaneous feature requests such as download count.

Milestone: All the requirements met

All specified work in Work Breakdown Structure (WBS) to be completed before Evaluation 2

Week 9-10: To Be Decided

Will be decided by discussing with mentor. Some alternatives are:

- Continue working on app store with additional functionality such as build desktop version of app store
- Work on porting one python 2 activity to python 3

Week 11 & 12: Buffer Time

Buffer tine for unexpected but likely delays or additional functionalities or modifications

Biography/Competency

I am studying Master of Computer Science (Software Engineering) from University of Wollongong, Australia. I am originally from commerce (business) background but due to my passion for programming and than gaining interest in broader field of software engineering and computer science, I decided to obtain formal education in it. I have done extensive programming in half a dozen programming languages for well over three years. I am well-experienced with technologies and knowledge which would be required in this project such as python, git, JavaScript, HTML5, MySQL, SSH, Nginx, requirements engineering, UX/UI design etc. Combining this with the fact that this project will be built from scratch with minimal interaction with current codebase ensures very high success possibility of the proposed project. Before this project, I made only non-code contributions to FLOSS projects (mainly OpenStreetMap) but all the code that I publish, I publish it under free software license which you can see on my website including Internet Media Manager app written in python. I am starting *code* contributions to Sugar Labs (and other FLOSS projects) which you can see on my newly created GitHub profile.

Post GSoC 2020

Once App Store is build, I intend to contribute in completing porting of Sugar desktop and activities to python3. By the end of GSoC 2020, by working on app store and in other activities of community such as in bug filling and fixing of Sugar core and activities, I will be a lot more familiar with codebase and more effectively work towards porting codebase. If members in community support the idea, than I also intend to make native app store for Sugar based on app store website.