Project Proposal
Google Summer of Code 2019

Ketki Salunkhe

Create a Sugar Dashboard

Sugarlabs

Basic Details

Full Name	Ketki Avinash Salunkhe
Institute	Atharva College of Engineering, Malad(west), Mumbai.
Degree	B.E (Computer Science)
Email	salunkheketki23@gmail.com
Github	https://github.com/salunkheketki19
First Language	Marathi
Location Timezone	Mumbai Indian Standard Time (GMT+0530)
Emergency Contact	Ruturaj Gujar ruturaj2305@gmail.com +91-9167152586

Your Motivation

What is your motivation to take part in Google Summer of Code?

I’ve always been passionate about writing code. The one thing I haven’t experienced yet is working with an open source organization. The most intriguing thing about open source is how people from various backgrounds work across different timelines without borders to achieve one common goal, the success of their project without expecting anything
in return.

Generally, I would’ve taken up a summer internship and worked at a startup that I found interesting. But in the end, it would still be just a regular monotonous internship. Once the duration of the internship is over, my association with the company would end and I would never know how (if?) my code was being used. Therefore, instead of the regular, I decided I would like to start contributing to an open source community and work with developers from around the world.

This is where the Google Summer of Code program steps in. Not only does the program provide an opportunity for an individual like me to contribute to open source projects but it also gives me the chance to be mentored by few of the most experienced contributors of the community. While I know a mentor isn’t a replacement to a Google or a stack overflow, it is nice knowing that someone is there to help you. One of my goals is to learn as much as possible from my assigned mentors as long as it isn’t much trouble to them. Additionally, GSoC serves as a stepping stone into the world of open source projects. This ensures that even after the end of the program I can still continue working with the organization and help in any way possible while at the same time seeing my code being used by others around the world.

Why do you choose Sugar Labs?

Sugarlabs is a community-run software project whose mission is to produce, distribute, and support the use of Sugar, an open source desktop environment and learning platform. Sugarlabs being a non-profitable organization gained my interest and eagerness to work with NGO. Working with an NGO will also help me pay back to society. I believe working with the mentors at Sugar labs will be an invaluable experience.I feel my skills and enthusiasm will be a great addition to the organization.

Why do you want to work on this particular project?

I want to work on Create a Sugar Dashboard as this project is completely based on python. I have prior experience working with python which helps me to code better. Also since I am new to GTK, I will be able to learn new stuff which will be helpful for me in future. In this project, I have to create a completely new application which will help me contribute to an NGO.

What are your expectations from us during and after successful completion of the program?

During the program, I expect to interact with the mentors about topics ranging from the issues that I am facing while trying to implement a particular feature to python and Gtk in general. With that said, I understand that the mentors are busy and I don’t consider them as a replacement to Google. Instead, I’d prefer to interact with them whenever they’re free just for some time. My main goal is to learn as much as I can over the summer.

Apart from working on the project, I expect to hone my python skills over and after the summer. I expect the mentors, who are quite experienced in the field to help me out with that and suggest a few resources if I’m stuck at something or am unable to grasp a concept. As this isn’t a part of the project, I wouldn’t want to take up too much of the mentor’s time and would hence ask them when(if?) they’re free to discuss this.

If selected, I would like to keep on contributing to Sugar Labs beyond the summer. The ultimate goal would be to be appointed as a mentor next year and help a young student start contributing to Sugar labs the same way I had received help. This would be my way of paying it forward and helping the community not only with my services this summer but throughout the year.

What technology you will be using?

I will be using Python, GTK.

Project Details

What are you making?

The task that I aim to complete is **Create a Sugar Dashboard**. The dashboard is an information management tool that helps to analyze and displays the performance of the users. The dashboard will be created for the users to view their latest activities performed, last projects opened etc which will be visualized through a heatmap. The dashboard will help admin and an individual to inspect all the activities performed by them.

Currently, the XO on the website displays the options on tap on which the user needs to click which in turn a opens a GTK widget on your desktop showing the menu. The task here is to design a widget that can substitute the XO in the center that gives the user the overall health condition of the computer such as battery status, disk space.

The next task is to create an app similar to the existing journal which could be modified by the users according to their requirements. The features that should be extended should help Parents, Teachers, Students, Sugar Labs community. Apart from this, the features to be added are the integration of portfolio activities and the dashboard.

How will it impact Sugar Labs?

This project will help Sugar Labs users to maintain the records of activities performed on the dashboard. It will help admin to maintain the record of the activities performed. It will also help the users to make modifications in the journal according to their requirement. It will help Sugar Labs to satisfy the demands of the users. It will enhance the user experience on the XO as the widget for displaying machine information will be available on the Desktop.

Project Goals

Tasks

1. Create a Sugar Dashboard

- The dashboard will appear containing the set of activities that can be performed by the user.
- The dashboard will be designed using Glade in GTK python3.
- Glade is a RAD tool to enable quick and easy development of user interfaces. The user interface saved in Glade are saved as XML by using GtkBuilder GTK+ object and can be loaded by application dynamically
- Matplotlib, which is an open source library to plot graphs, can be used to plot heatmaps. It can be embedded with GTK3.
- The heatmaps will be displayed on the dashboard. It will contain different colors representing different activities done during the entire day.
- The user activities performed will be stored in the database.

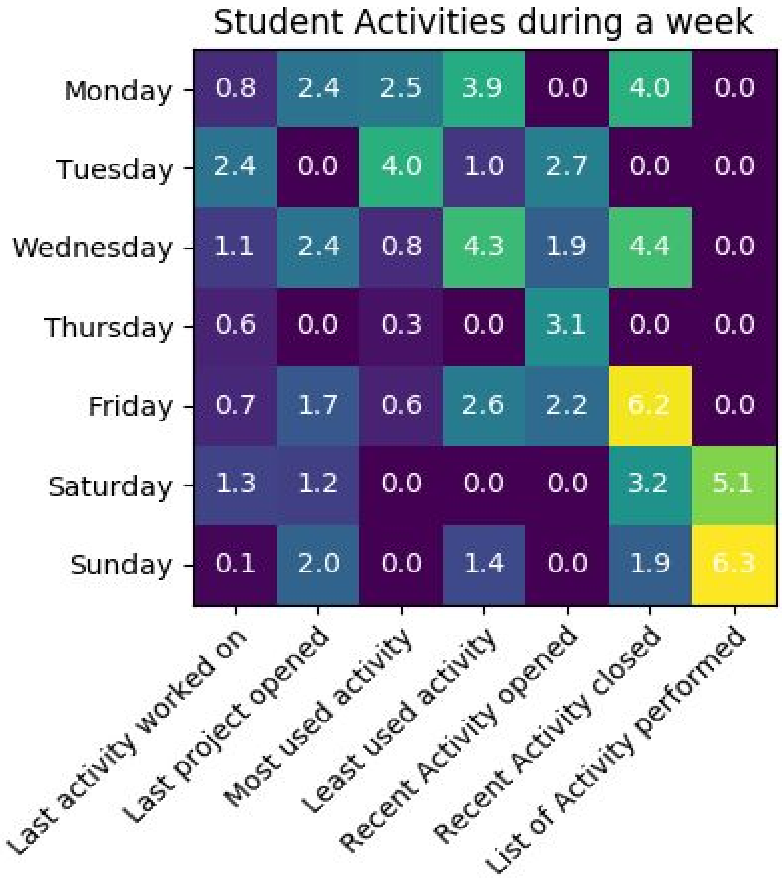
- The Matplotlib library will take the input to produce real-time graphs of different activities.
- The panel to the left will display the list of activities like the least opened activity, last activity opened, recent activities, etc.
- When the user will click on any of the option present on the side panel, a bar graph will appear indicating the corresponding activities performed.

This is a sample output for the heatmap.

```
import matplotlib
matplotlib.use('GTK3Agg')  # or 'GTK3Cairo'
import matplotlib.pyplot as plt
import numpy as np
from matplotlib.colors import LogNorm

day                                                                    =
["Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sund
ay"]
activity = ["Last activity worked on","Last project opened","Most
used activity","Least used activity","Recent Activity opened",
"Recent Activity closed","List of Activity performed"]

student = np.array([[0.8, 2.4, 2.5, 3.9, 0.0, 4.0, 0.0],
                    [2.4, 0.0, 4.0, 1.0, 2.7, 0.0, 0.0],
                    [1.1, 2.4, 0.8, 4.3, 1.9, 4.4, 0.0],
                    [0.6, 0.0, 0.3, 0.0, 3.1, 0.0, 0.0],
                    [0.7, 1.7, 0.6, 2.6, 2.2, 6.2, 0.0],
                    [1.3, 1.2, 0.0, 0.0, 0.0, 3.2, 5.1],
                    [0.1, 2.0, 0.0, 1.4, 0.0, 1.9, 6.3]])



fig, ax = plt.subplots()
im = ax.imshow(student)

# We want to show all ticks...
ax.set_xticks(np.arange(len(activity)))
ax.set_yticks(np.arange(len(day)))
# ... and label them with the respective list entries
ax.set_xticklabels(activity)
ax.set_yticklabels(day)
```

```
# Rotate the tick labels and set their alignment.
plt.setp(ax.get_xticklabels(), rotation=45, ha="right",
       rotation_mode="anchor")

# Loop over data dimensions and create text annotations.
for i in range(len(activity)):
    for j in range(len(day)):
       text = ax.text(j, i, student[i, j],
                   ha="center", va="center", color="w")

ax.set_title("Student Activities during a week")
fig.tight_layout()

manager = plt.get_current_fig_manager()
# you can also access the window or vbox attributes this way
toolbar = manager.toolbar

# now let's add a button to the toolbar
import gi
gi.require_version('Gtk', '3.0')
from gi.repository import Gtk
pos = 8  # where to insert this in the mpl toolbar


# now let's add a widget to the vbox
label = Gtk.Label()
label.set_markup('Drag mouse over axes for position')
label.show()
vbox = manager.vbox
vbox.pack_start(label, False, False, 0)
vbox.reorder_child(manager.toolbar, -1)


def update(event):
    if event.xdata is None:
       label.set_markup('Drag mouse over axes for position')
    else:
        label.set_markup('<span color="#ef0000">x,y=(%f, %f)</span>'
% (event.xdata, event.ydata))
```

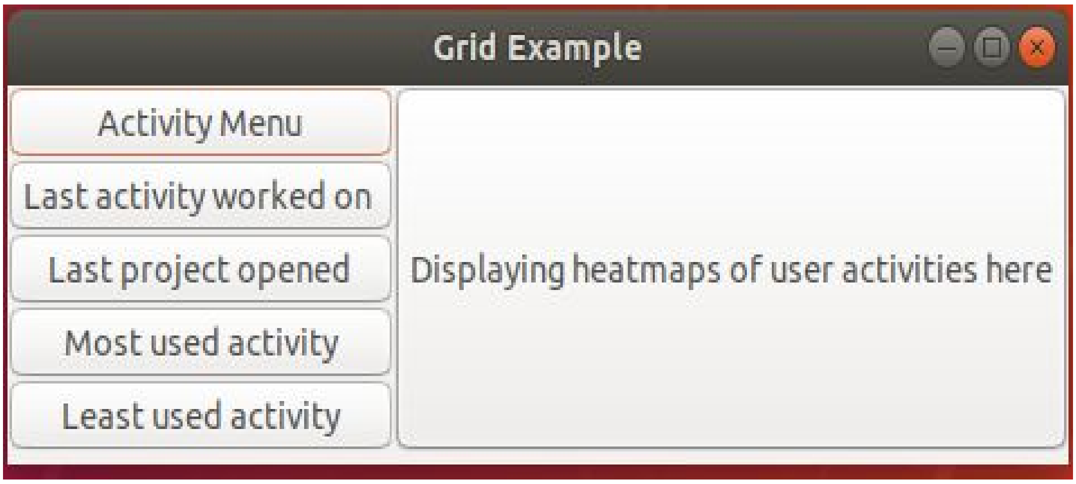
```
plt.connect('motion_notify_event', update)
plt.show()
```



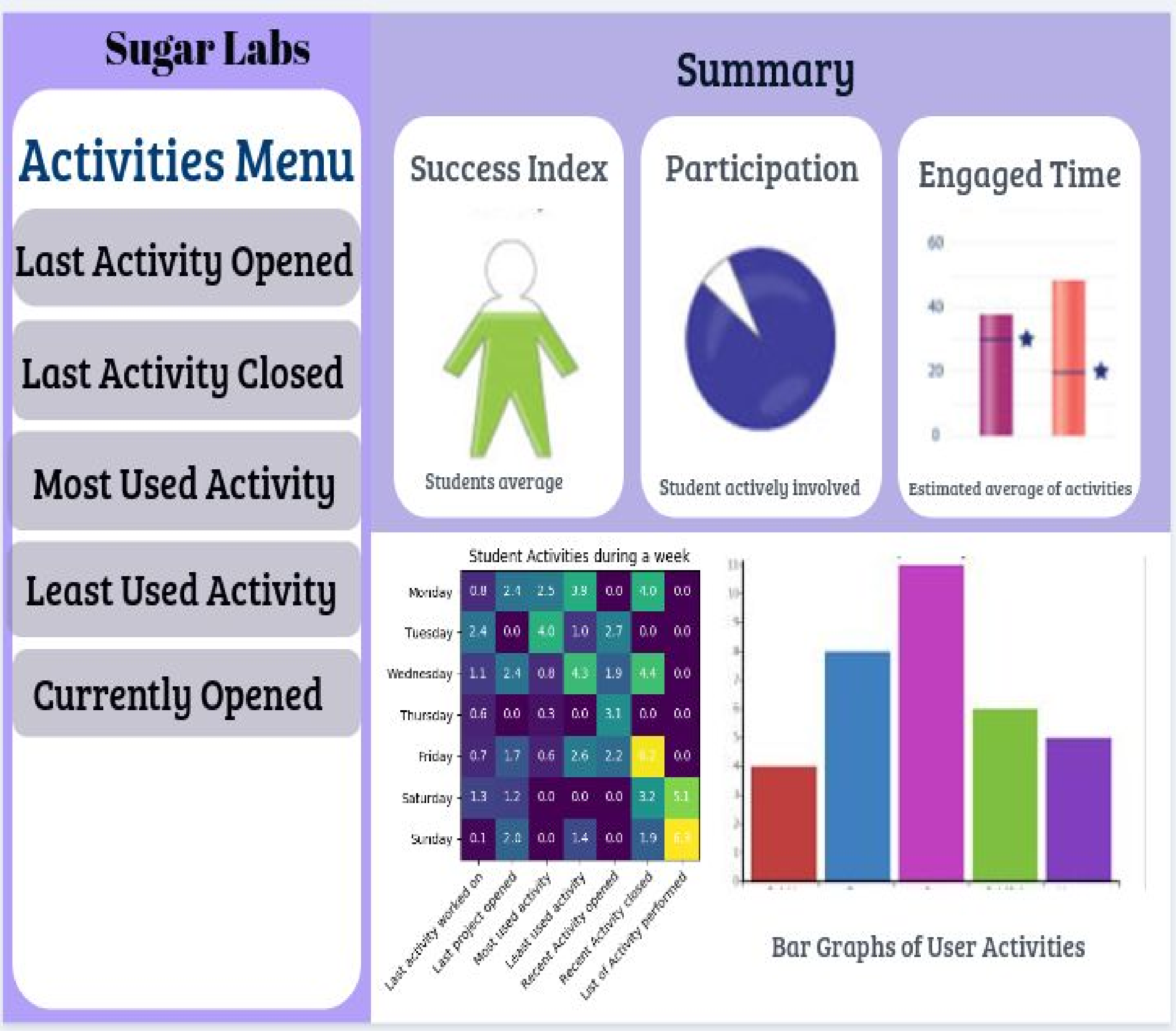
The below-mentioned code displays an idea for dashboard

```
import gi
gi.require_version('Gtk', '3.0')
from gi.repository import Gtk

class GridWindow(Gtk.Window):

    def __init__(self):
        Gtk.Window.__init__(self, title="Grid Example")
        self.set_default_size(150, 150)
        grid = Gtk.Grid()
        self.add(grid)


        button1 = Gtk.Button(label="Activity Menu")
```

```
            button2  =  Gtk.Button(label="Displaying  heatmaps  of  user
activities here")
        button3 = Gtk.Button(label="Last activity worked on ")
        button4 = Gtk.Button(label="Last project opened")
        button5 = Gtk.Button(label="Most used activity")
        button6 = Gtk.Button(label="Least used activity")

        grid.add(button1)
        grid.attach(button2, 10, 0, 1000, 1000)
                                 grid.attach_next_to(button3,    button1,
Gtk.PositionType.BOTTOM, 1, 1)
                                 grid.attach_next_to(button4,    button3,
Gtk.PositionType.BOTTOM, 1, 1)
        grid.attach(button5, 0, 4, 1, 1)
                                 grid.attach_next_to(button6,    button5,
Gtk.PositionType.BOTTOM, 1, 1)

win = GridWindow()
win.connect("destroy", Gtk.main_quit)
win.show_all()
Gtk.main()
```



Sample Design for Sugar Dashboard. This dashboard is designed for kids to attract them with different colors.

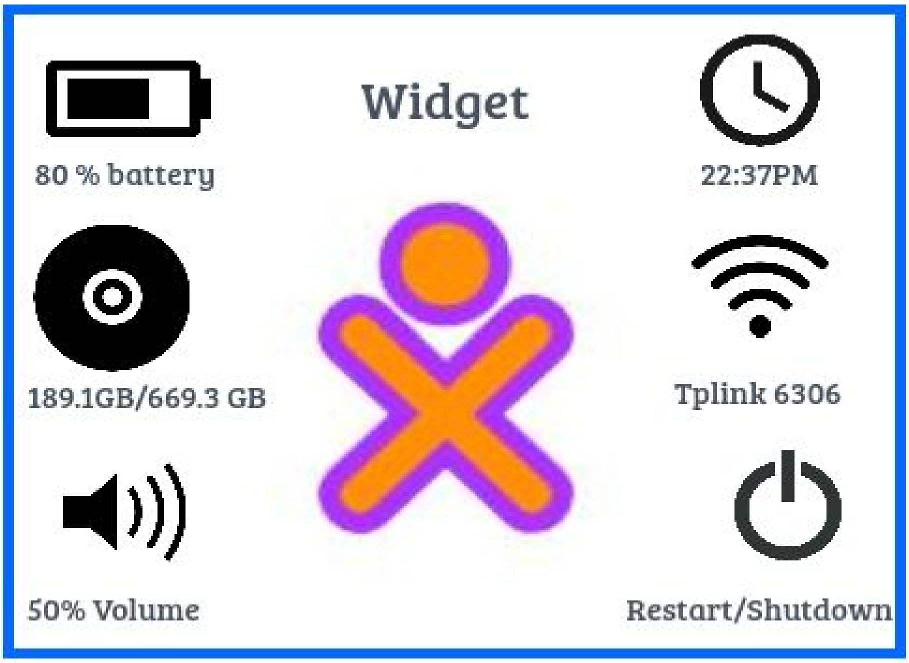


2. Tamagotchi

- .The task here is to replace the existing XO with a widget without losing any of its functionalities.
- The widget will be designed using GTK. GTK is a multi-platform toolkit to create a user interface.
- When the user clicks on the widget present on the website it will be opened on the users desktop.
- The user can then click on the options available such as battery status, disk space available, language, security.
- The icons available on the widget will display the information at-a-glance. The user will not need to go to

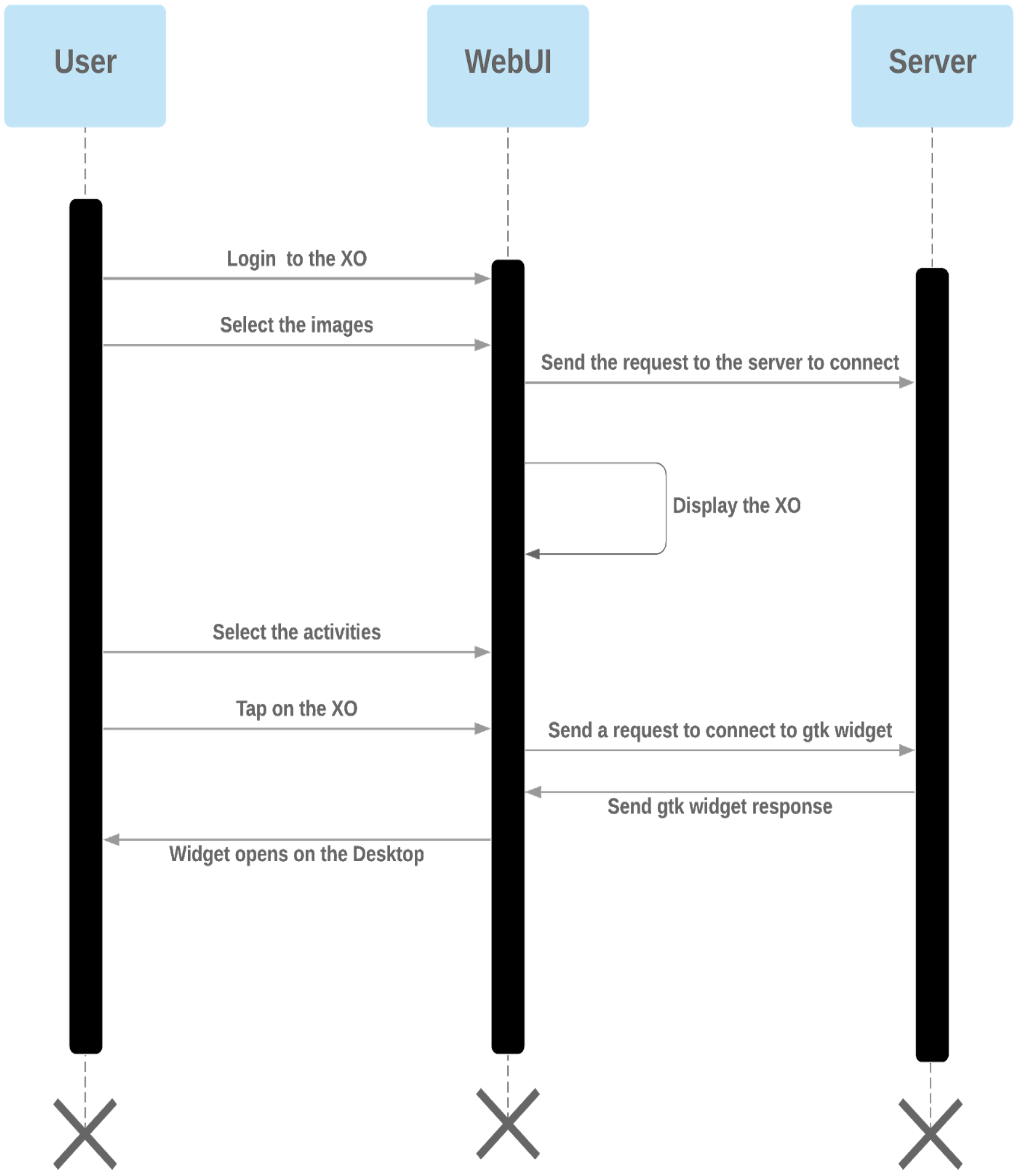
the different panel and get the task done, thus providing ease to the user

Design for XO Widget



Code for battery status and Disk space taken for codebase of Sugar Labs.

```
def _battery_info(self, item):
    root = '/sys/class/power_supply/olpc-battery/'
    if not os.path.exists(root + item):
        return ''

    return self.__read_file(root + item).strip()


def battery_capacity(self):
       return self._battery_info('capacity') + ' ' + \
           self._battery_info('capacity_level')

   def battery_info(self):
       # Should be just:
       # return self._battery_info('uevent')
         # But because of a bug in the kernel, that has trash, lets
filter:
       bi = ''
       for line in self._battery_info('uevent').splitlines():
           if line.startswith('POWER_'):
               bi += line + '\n'
       return bi

   def disksize(self, path):
       return os.statvfs(path).f_bsize * os.statvfs(path).f_blocks

   def diskfree(self, path):
       return os.statvfs(path).f_bsize * os.statvfs(path).f_bavail
```

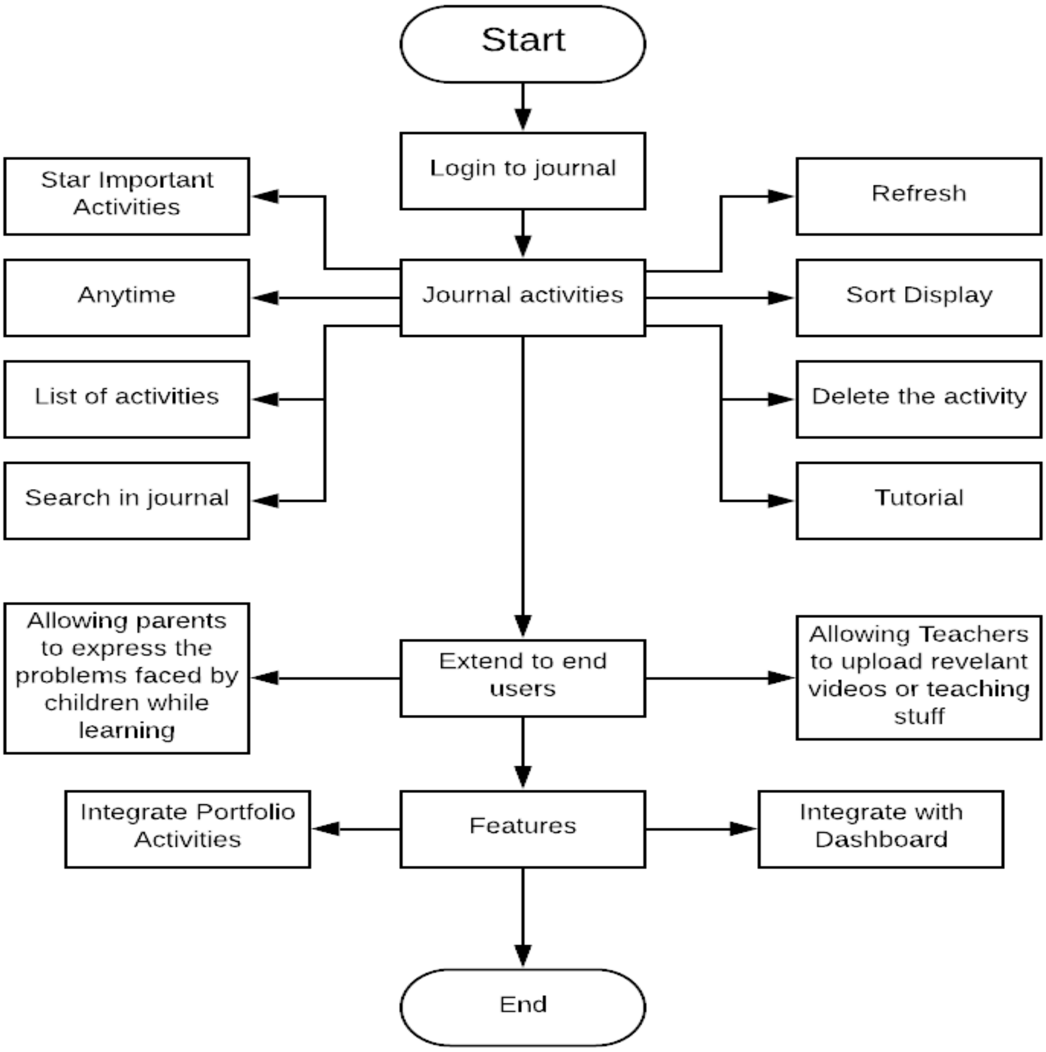


Basic sequence diagram for Tamagotchi

3. Journal

- The task here is to create an application similar to the existing journal and extend it to the end users
- The idea here is to allow parents to express the difficulties faced by their child so that Sugar Labs can create such activities that will help the young mind to grab faster.

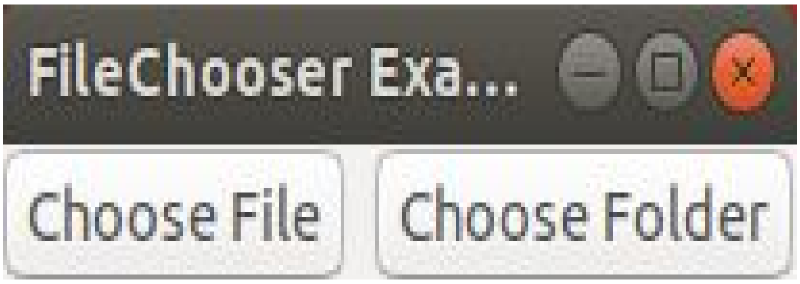
- Also, we can allow teachers to upload good videos for the students that will help them to learn faster, upload teaching stuff such as homework in the form of fun activities.
- If the application allows the above-mentioned solutions it will be beneficial to parents, teachers, students as well as Sugar Labs community.
- The application should be then integrated with portfolio activities and dashboard mentioned above.


Start
Login to journal
Star Important Activities
Refresh
Journal activities
Anytime
Sort Display
List of activities
Delete the activity
Search in journal
Tutorial
Allowing parents to express the problems faced by children while learning
Extend to end users
Allowing Teachers to upload revelant videos or teaching stuff
Integrate Portfolio Activities
Features
Integrate with Dashboard
End

Code to select file/folder from the desktop for teachers to upload

```
import gi
gi.require_version('Gtk', '3.0')
from gi.repository import Gtk

class FileChooserWindow(Gtk.Window):

    def __init__(self):
       Gtk.Window.__init__(self, title="FileChooser Example")

       box = Gtk.Box(spacing=6)
       self.add(box)

       button1 = Gtk.Button("Choose File")
       button1.connect("clicked", self.on_file_clicked)
       box.add(button1)

       button2 = Gtk.Button("Choose Folder")
       button2.connect("clicked", self.on_folder_clicked)
       box.add(button2)

    def on_file_clicked(self, widget):
       dialog = Gtk.FileChooserDialog("Please choose a file", self,
           Gtk.FileChooserAction.OPEN,
           (Gtk.STOCK_CANCEL, Gtk.ResponseType.CANCEL,
           Gtk.STOCK_OPEN, Gtk.ResponseType.OK))

       self.add_filters(dialog)

       response = dialog.run()
       if response == Gtk.ResponseType.OK:
           print("Open clicked")
           print("File selected: " + dialog.get_filename())
       elif response == Gtk.ResponseType.CANCEL:
           print("Cancel clicked")

       dialog.destroy()

    def add_filters(self, dialog):
       filter_text = Gtk.FileFilter()
       filter_text.set_name("Text files")
       filter_text.add_mime_type("text/plain")
```

```
        dialog.add_filter(filter_text)

        filter_py = Gtk.FileFilter()
        filter_py.set_name("Python files")
        filter_py.add_mime_type("text/x-python")
        dialog.add_filter(filter_py)

        filter_any = Gtk.FileFilter()
        filter_any.set_name("Any files")
        filter_any.add_pattern("*")
        dialog.add_filter(filter_any)

    def on_folder_clicked(self, widget):
           dialog = Gtk.FileChooserDialog("Please choose a folder",
self,
            Gtk.FileChooserAction.SELECT_FOLDER,
            (Gtk.STOCK_CANCEL, Gtk.ResponseType.CANCEL,
            "Select", Gtk.ResponseType.OK))
        dialog.set_default_size(800, 400)

        response = dialog.run()
        if response == Gtk.ResponseType.OK:
            print("Select clicked")
            print("Folder selected: " + dialog.get_filename())
        elif response == Gtk.ResponseType.CANCEL:
            print("Cancel clicked")

        dialog.destroy()

win = FileChooserWindow()
win.connect("destroy", Gtk.main_quit)
win.show_all()
Gtk.main()
```



Timeline

Community Bonding period (May 6- May 26)

During the community bonding period, I aim to go through the entire Sugarlabs codebase thoroughly. Currently, I have enough knowledge to add or modify some features. However, going through the entire codebase will help me to work faster during my coding period.

I would like to interact with the mentors as well as get to know the community as a whole and helping out whenever I can.

I am currently trying to sharpen my skills on python and javascript. I expect to give it my best right from the start of the community bonding period instead of waiting for the coding period to start.

Week 1 (May 27 - June 3)

- Verify the design of the dashboard with the mentors.
- Implementation of the designed dashboard.
- Implement the activities performed by the users such as last activity opened, the last project worked on by producing heatmap.

Week 2(June 3 - June 10)

- Continue with the implementation of the dashboard for the least used activity, most used activity.
- Highlighting the feature of the dashboard.
- Complete the implementation of the dashboard.

Week 3(June 11 - June 17)

- Test the application.
- Begin with task 2 to design a widget to replace an XO

Week 4(June 18 - June 24)

- Time period for any unexpected delay.

June 24 (First Evaluation Phase)

Deliverables

- Design of the dashboard.
- Complete implementation of the dashboard.
- Design of the widget.

Week 5(June 25 - July 7)

- Providing the overall health condition of the computer such as battery status, available disk space etc on hover on the widget.

Week 6(July 8 - July 14)

- Testing the implementation of the XO.
- Adding additional features if required.

Week 7(July 16 - July 22)

- Time period for any unexpected delay.

July 22(Second Evaluation Phase)

Deliverables

- Complete implementation of the widget

Week 8(July 23- July 29)

- Workflow of the app similar to a journal.
- Implementation of the features of the journal.

Week 9(July 30 - August 7)

- Continue the implementation of features for the new journal.
- Implement the functionality for the user to allow extending the application according to their requirement.

Week 10(August 7 - August 13)

- Testing the application.
- Integrating the developed application with portfolio activities and dashboard which was designed earlier.

Week 11(August 14 - August 19)

- Time period for any unexpected delay.

August 19(Final Evaluation Phase)

Deliverables

- Complete the application with all necessary features and frontend.

How much time will I be able to contribute to this project?

I will be working for 40-45 hours a week.

After GSoC, ends I plan on continuing my work with Sugar Labs and help on improving UI for the dashboard. As has been stated earlier in my proposal, my goal is to get so good that I can mentor a student next year. That would be my way of paying back to the community which gave me such an excellent opportunity. I would like to work on other project with Sugar Labs.