

BalanceBuddy

James Zesiger, Preston Speidel, Connor Valley

Intro (names/major/project role, process model)

- James Zesiger
 - Computer Science Major
 - Role: Developer, Quality Assurance
- Preston Speidel
 - Computer Science Major
 - Role: Developer, Analyst
- Connor Valley
 - Computer Science Major
 - Role: Developer, Project Lead
- Agile: Scrum



What is BalanceBuddy?

Desktop budgeting application for college students.

Essential Requirements:

- Application shall allow users to create new profiles and sign into existing profiles
- User profiles shall store users tracking information
- Users shall be able to track expenses and income

Desirable Requirements:

- Users shall be rewarded for using the application with levels and experience system

Optional Requirements:

- Users shall be able to log loan payments

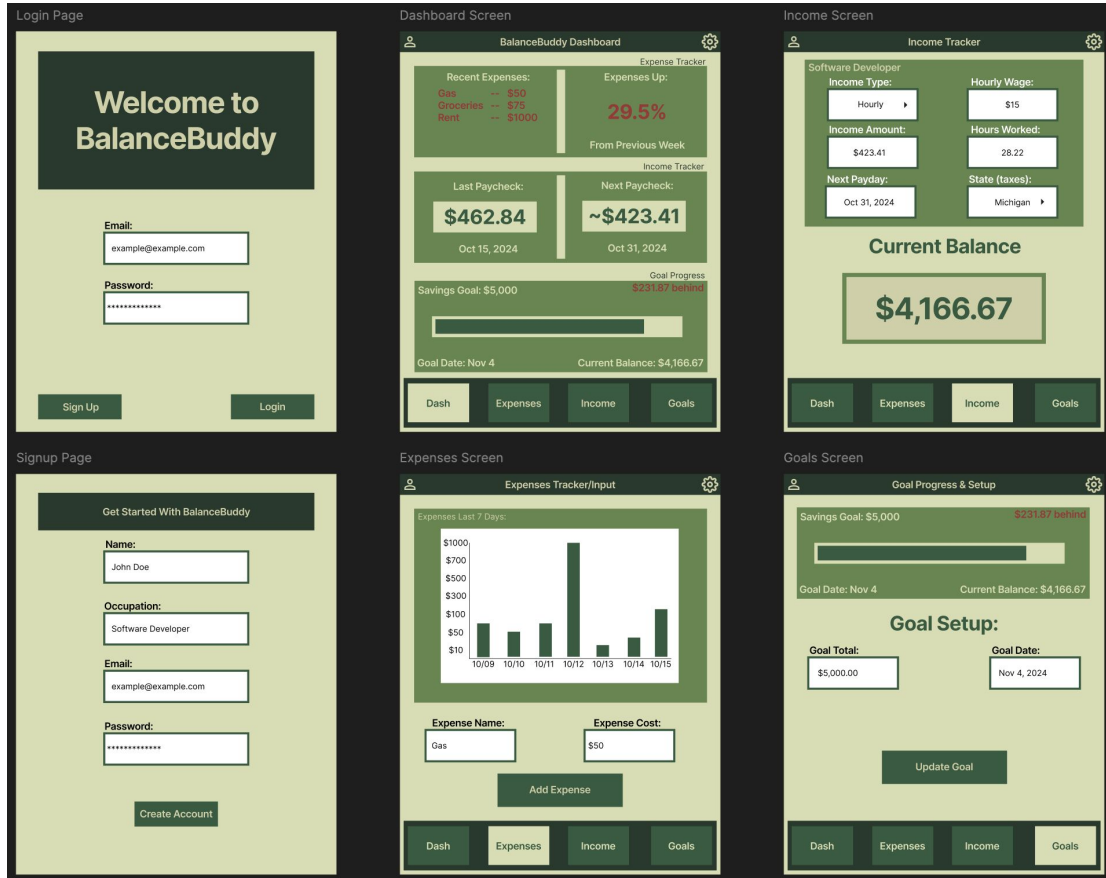


Timeline

[illegible]

What we've done so far

GUI Mockup (Figma)



GUI in Python

```
src > gui.py > ...
1 import pygame
2 import sys
3
4 # initialize pygame instance
5 pygame.init()
6
7 # setup window size and title
8 width, height = 768, 960
9 screen = pygame.display.set_mode((width,height))
10 pygame.display.set_caption("Budgeting App GUI Mockup")
11
12 # setup colors for screen elements
13 dock_color = (37, 58, 45)
14 dock_option_color = (48, 91, 63)
15 module_color = (97, 135, 76)
16 text_color = (207, 207, 165)
17 background_color = (215, 221, 176)
18
19 # setup fonts
20 pygame.font.init()
21 header_font = pygame.font.Font(None, 36)
22
23 # setup text
24 header_text = header_font.render("Welcome to Your Budget Dashboard!", True, text_color)
25
26 # setup images
27 user_icon = pygame.image.load("images/user_icon.png")
28 user_icon = pygame.transform.scale(user_icon, (40,40))
29 settings_icon = pygame.image.load("images/settings_icon.png")
30 settings_icon = pygame.transform.scale(settings_icon, (40,40))
31
32 # main app loop
33 running = True
34 while running:
35     for event in pygame.event.get():
36         if event.type == pygame.QUIT:
37             running = False
38
39     # fill in background color
40     screen.fill(background_color)
41
42     # setup header element
43     pygame.draw.rect(screen, dock_color, (0, 0, width, 50))
44     screen.blit(header_text, (170, 15))
45     screen.blit(user_icon, (10,5))
46     screen.blit(settings_icon, (width-50, 5))
47
```

User-Profiles Class

```
src > User_Profile.py > User
1 class User:
2     def __init__(self, username, email, password):
3         self.__username = username
4         self.__email = email
5         self.__password = password
6         self.__expenses = {}
7         self.__experience = 0
8
9     @property
10     def username(self):
11         return self.__username
12
13     @username.setter
14     def username(self, new_username):
15         self.__username = new_username
16
17     @property
18     def email(self):
19         return self.__email
20
21     @email.setter
22     def email(self, new_email):
23         self.__email = new_email
24
25     @property
26     def password(self):
27         return self.__password
28
29     @password.setter
30     def password(self, new_password):
31         self.__password = new_password
32
33     @property
34     def expenses(self):
35         return self.__expenses
36
37     @expenses.setter
38     def expenses(self, new_expenses):
39         self.__expenses = new_expenses
40
41     def add_expenses(self, date, expense):
42         self.__expenses[date] = expense
43
44     def remove_expenses(self, date):
45         del self.__expenses[date]
46
47     @property
48     def __experience(self):
49         return self.__experience
50
```