

The Four Corners presents:

The logo for 'Degree Dollar\$' is centered within a light green rounded rectangle, which is itself set against a darker green background. The text 'Degree' is in a bold, black, sans-serif font. Below it, 'Dollar\$' is in a larger, bold, black, sans-serif font. To the right of the text is a stylized illustration of a green banknote with three white circles and a black graduation cap (mortarboard) positioned as if it is about to land on or has just landed on the note.

Degree Dollar\$

A budgeting application for college students



We Are: The Four Corners

Jason Tanis

- **Major:** Statistics
- **Minor:** Computer Science
- **Roles:** Project Manager, Python Developer, Database Manager, Android Tester

Sam Bergman

- **Majors:** Computer Science, Piano Performance
- **Roles:** Python Developer, Assistant Database Manager

Kelsey Tedford

- **Majors:** Mathematics (Applied Emphasis), Statistics
- **Minor:** Data Science
- **Roles:** Prototype Developer, Assistant Database Manager, iOS Tester

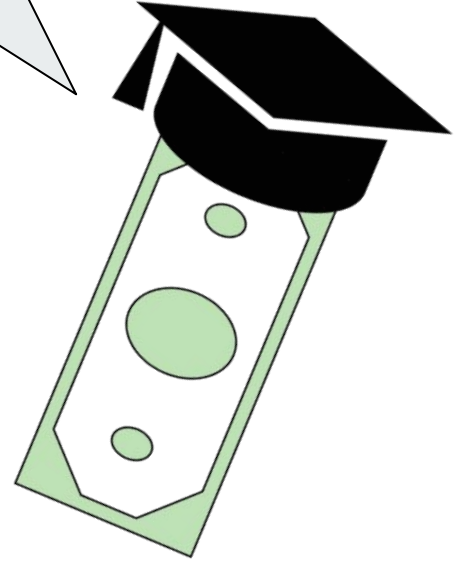
Tony Choummanivong

- **Major:** Cybersecurity
- **Roles:** Python Developer, App Tester

Why Degree Dollars?

- Designed for undergraduate and graduate college students
 - Students incur a lot of expenses
- Degree Dollars is a *budgeting* app
 - Can plan for expenses and keep consistent record of available funds
- Primarily developing for Android/iOS
 - Desktop version will also be available

Be a savings scholar!!



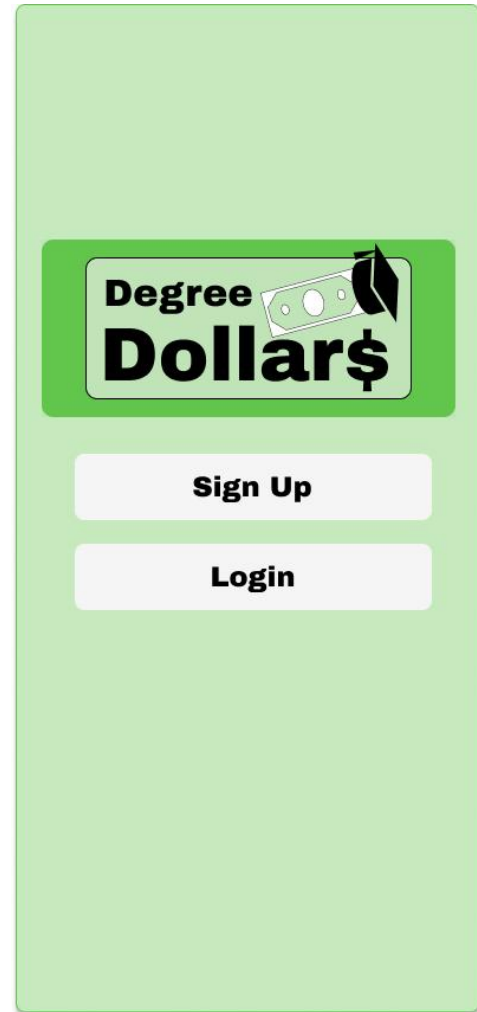


Functionality Overview

- Login
- Budget
- Expense & Income
- History
- Loan Calculator

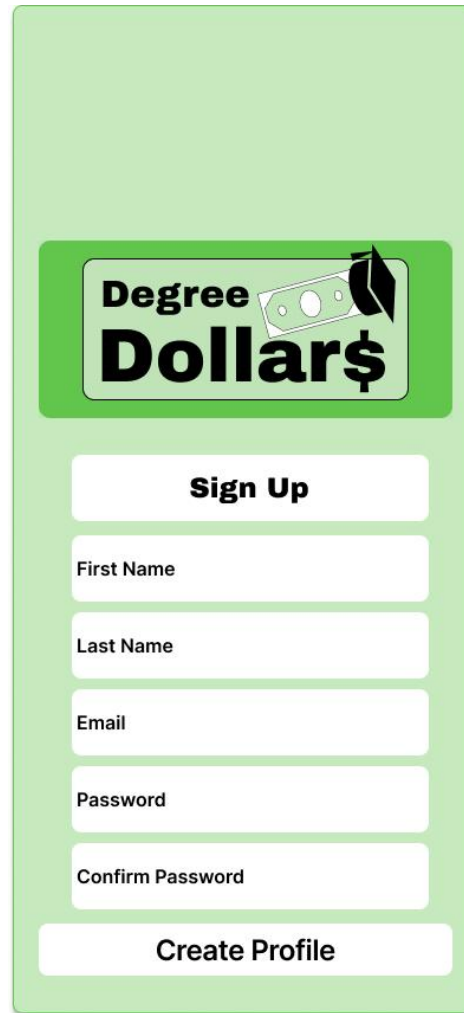
Login

- Not yet implemented in BeeWare
- Currently, just a “View my Budgets” button (left)
- See Figma prototype (right)

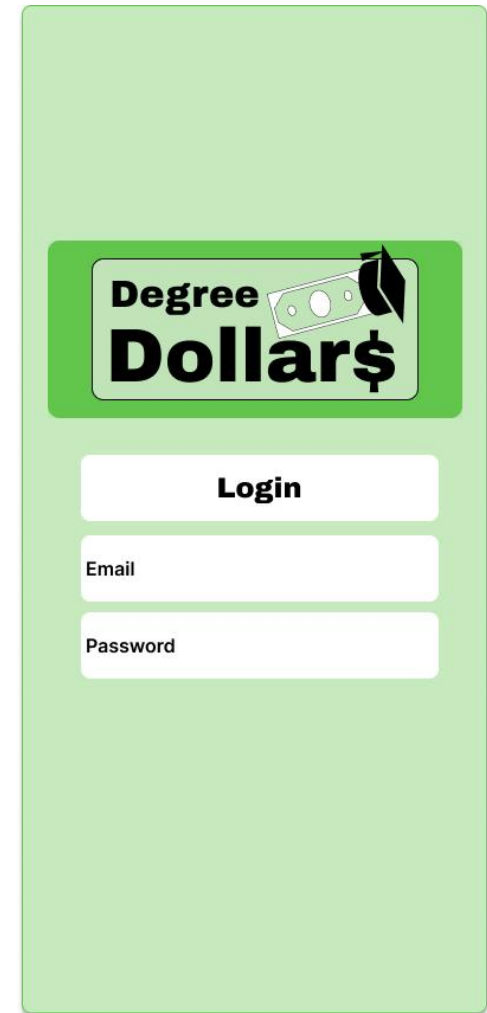


Login (cont.)

- “Sign Up” and “Login” features have been prototyped in Figma
 - “Sign Up” to create new account
 - “Login” for existing accounts
- Having a user account would allow for app use across multiple devices (e.g., phone and computer)
 - Otherwise, user activity could only be locally saved
- This will be one of our last features to implement



A mobile app prototype for the 'Sign Up' feature. At the top is a green header with the text 'Degree Dollar\$' and a graphic of a dollar bill. Below the header is a white button labeled 'Sign Up'. Underneath the button are five white input fields with labels: 'First Name', 'Last Name', 'Email', 'Password', and 'Confirm Password'. At the bottom is a white button labeled 'Create Profile'.



A mobile app prototype for the 'Login' feature. At the top is a green header with the text 'Degree Dollar\$' and a graphic of a dollar bill. Below the header is a white button labeled 'Login'. Underneath the button are two white input fields with labels: 'Email' and 'Password'.

Budget

- In BeeWare, budgets cannot yet be saved
 - “Save Budget” only returns user to Home Screen
- Additionally, we still need to allow user to:
 - Add their own categories
 - Add additional subcategories

3:37 9

Degree Dollars

Create New Budget

Indicate current month 2

Food

Groceries

300.00

Transportation

Gas

50.00

Education

Textbooks

200.00

Entertainment

Subcategory

0.00

SAVE BUDGET

{Month}'s Budget

{Section}

{Subsection}

\$(Budget - Expense) left

\$(Amount)

{Section}

{Subsection}

\$(Budget - Expense) left

\$(Amount)

{Section}

{Subsection}

\$(Budget - Expense) left

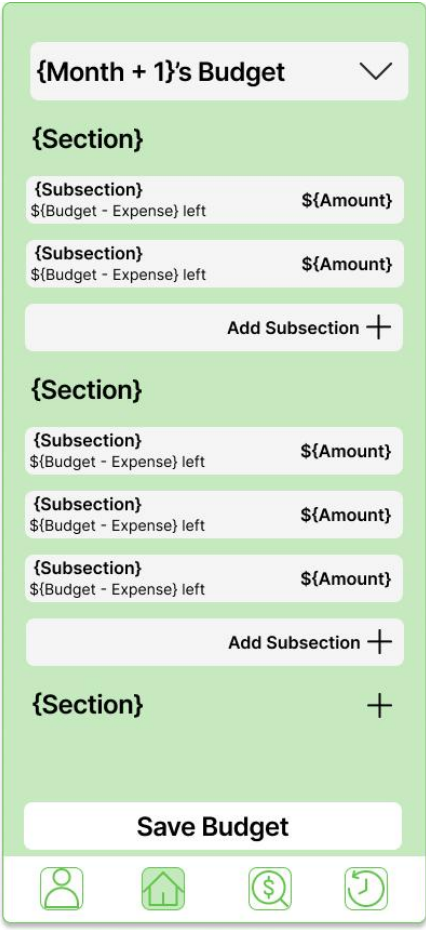
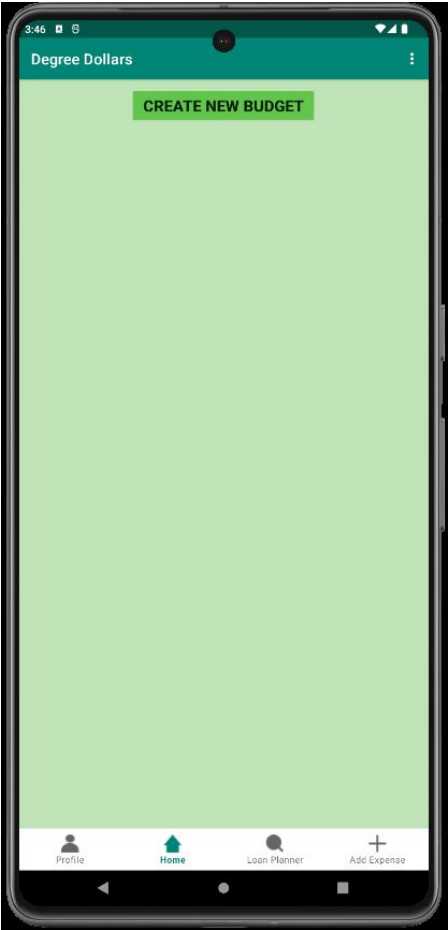
\$(Amount)

+

Person, Home, Dollar, Clock icons

Budget (cont.)

- After saving their first budget, user can add budgets for remaining months of current year
 - This has not yet been implemented in BeeWare
 - Only have navigation bar and “Create New Budget” button



Expense & Income

- Not yet implemented in BeeWare
- Allows user to add/subtract money from a subcategory of their choosing

{Month}'s Budget

{Section}

{Subsection}

\$(Budget - Expense) left

\$(Amount)

{Subsection}

\$(Budget - Expense) left

\$(Amount)

{Section}

{Subsection}

\$(Budget - Expense) left

\$(Amount)

{Subsection}

\$(Budget - Expense) left

\$(Amount)

ExpenseIncome

Date

{Date}

Amount

\$(Amount:.2f)

Section

{Subsection}

Merchant

{Merchant}

{Month}'s Budget

{Section}

{Subsection}

{Budget - Expense} left

{Amount}

{Subsection}

{Budget - Expense} left

{Amount}

{Section}

{Subsection}

{Budget - Expense} left

{Amount}

{Subsection}

{Budget - Expense} left

{Amount}

Expense

Income

Date

{Date}

Amount

\$(Amount:.2f)

Section

{Subsection}

Merchant

{Merchant}

{Month}'s Budget

{Section}

{Subsection}

{Budget - Expense} left

{Amount}

{Subsection}

{Budget - Expense} left

{Amount}

{Section}

{Subsection}

{Budget - Expense} left

{Amount}

{Subsection}

{Budget - Expense} left

{Amount}

Expense

{Subsection}

Date

{Subsection}

Amount

{Subsection}

Section

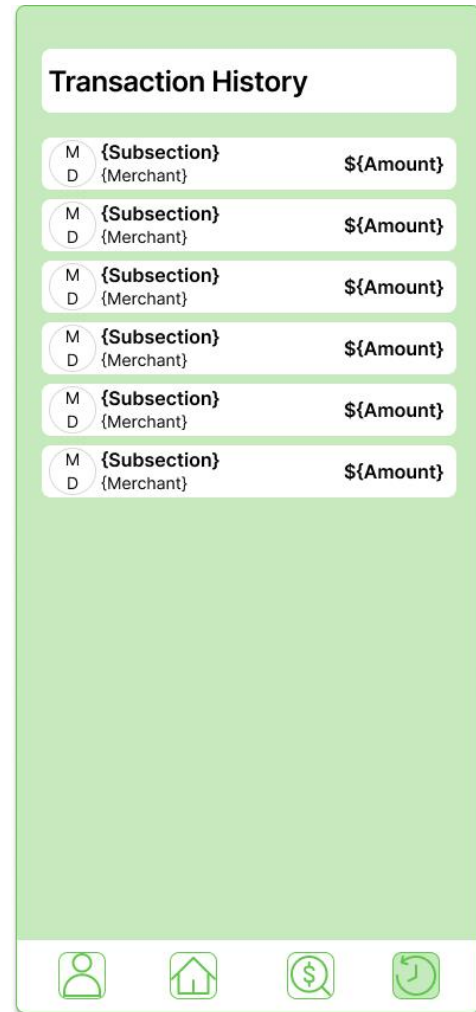
{Subsection}

Merchant

{Merchant}

History

- Not yet implemented in BeeWare
- Allows user to view all previous transactions (i.e., expense/income)





Loan Calculator

- Not yet prototyped or implemented
- Accepts multiple user values:
 - Total loan amount
 - Interest rate
 - Either:
 - Planned monthly payment
 - Planned number of months to have loan paid in full by
- Will calculate either:
 - Monthly payment to make
 - Number of months to pay loan in full by
- This will be one of our last features to work on

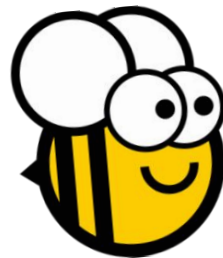
The Tech Behind Degree Dollars

The BeeWare suite:

- Briefcase (for app deployment on Android, iOS, and Desktop)
- Toga (for GUI of app)
- BeeWare uses Python code
 - All teammates familiar with Python
 - Easily understandable

SQLite

- Allows for connection between Python code and SQL/NoSQL databases
 - Databases to store user login info and saved budgets





Progress Summary

- Have had 5 SCRUM meetings, 4 with TA involved
- Team is progressing via regular prototyping and development
 - Communicating via Slack
- Currently implementing:
 - Budget
 - Expense & Income
- Nearly all prototyping is complete
 - See Done items

IN PROGRESS 2	DONE 9 ✓	
<div>Budgeting feature (BeeWare) 📅 06 MAR ✓ ECS-15</div>	<div>BeeWare Tutorial 📅 11 FEB ✓ ECS-10</div>	<div>Income Feature (Figma) 📅 06 MAR ✓ ECS-20</div>
<div>Expenses/Income feature (BeeWare) 📅 20 MAR ✓ ECS-17</div>	<div>Basic layout (Figma) 📅 11 FEB ✓ ECS-13</div>	<div>See my History Feature (Figma) 📅 27 MAR ✓ ECS-24</div>
	<div>UI/UX: Designing the app 📅 11 FEB ✓ ECS-9</div>	<div>User Login Feature (Figma) 📅 16 APR ✓ ECS-14</div>
	<div>Login Screen (BeeWare) 📅 18 FEB ✓ ECS-11</div>	
	<div>Budgeting Feature (Figma) 📅 20 FEB ✓ ECS-16</div>	
	<div>Spending Feature (Figma) 📅 06 MAR ✓ ECS-19</div>	



Progress Summary (cont.)

- To Do items primarily consist of:
 - Integrating SQL and NoSQL databases
 - Implementing features in BeeWare
- To summarize:

Total Jira tickets	19
To Do items (count, %)	8, 42.105%
In-Progress / Done items (count, %)	11, 57.895%
Done items (count, %)	9, 47.368%

TO DO 8

Integrating SQL Database with Python Script

01 APR

✓ ECS-29

Integrating NoSQL with Python Script

01 APR

✓ ECS-30

See my History Feature (BeeWare)

03 APR

✓ ECS-23

Loan Payment Plan Feature (BeeWare)

10 APR

✓ ECS-25

Loan Payment Plan Feature (Figma)

03 APR

✓ ECS-26

Deployment of app

16 APR

✓ ECS-27

Final Presentation

16 APR

✓ ECS-28

User Login Feature (BeeWare)

16 APR

✓ ECS-12

Gantt Chart

- Majority of prototype development is complete
- Implementation roughly 30% complete
 - Progress will likely speed up as development team becomes more efficient in coding
- Database integration to begin end of Feb./beginning of Mar.

TASK	Feb.-Apr. 2025		
	Feb.	Mar.	Apr.
Prototype Development (Figma)			
Basic layout (fitting it to a phone screen)			
UI/UX (designing colors, icons, and logo)			
Budgeting feature (user creating budgets)			
Spending feature (user subtracting money)			
Income feature (user adding money)			
See my history (list of previous user actions)			
Loan payment plan feature			
User login feature (sign up/log in) - not essential			
Implementation (BeeWare)			
BeeWare tutorial			
Opening screen (with logo)			
Budgeting feature (user creating budgets)			
Income/Expense feature (+/- from existing budget)			
See my history			
Loan payment plan feature			
User login feature - not essential			
Database Integration			
Integrate SQL database (to save user login info and transactions (i.e., "+/-"))			
Integrate NoSQL database (to save user budgets)			
Final Steps			
Final presentation preparation			
Final deployment of app			