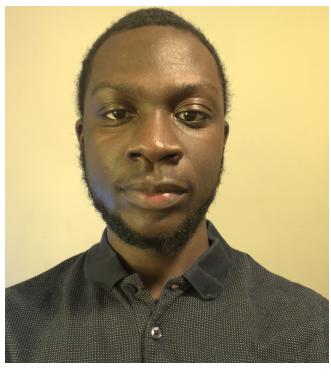


# #SP-25 Blue— Student Budgeting App

CS4850-03. Fall 2023,  
November 28, 2023,  
Professor Perry

## Team Members

	
Elliot Larez - Developer <a href="mailto:elliotlarez02@gmail.com">elliotlarez02@gmail.com</a> 470-707-7841	Jade McAlister- Developer, UX Designer <a href="mailto:mcalister.vjade@gmail.com">mcalister.vjade@gmail.com</a> 404-453-5623
	
Tori Porter - Project Design & Management <a href="mailto:victoriaanneporter@gmail.com">victoriaanneporter@gmail.com</a> 770.576.0626	Reginald Appiah - Developer, UX <a href="mailto:reginald.appiah7@gmail.com">reginald.appiah7@gmail.com</a> 4709638842

## Website

<https://sp25-blue.github.io/StudentBudgetApp/Website/index.html>

## GitHub

<https://github.com/SP25-Blue/StudentBudgetApp>

## I. Introduction

Mastering the art of budgeting proves to be a challenging skill for many individuals. Juggling various financial commitments such as bills, tuition, subscriptions, and daily expenses often creates a complex landscape that can seem nearly insurmountable. SpendSense endeavors to simplify this intricate process by offering swift and user-friendly methods to monitor spending, ensuring that each paycheck is optimized to its fullest potential.

SpendSense is a standalone financial tracking application designed for individuals. It is not part of a larger system. It will be developed using React Native, designed to run on multiple platforms. It enables users to track monthly income and expenses, categorize spending, set financial goals, and receive alerts for budget overspending.

SpendSense goes beyond merely tracking user spending; it serves as a comprehensive guide and resource hub to enhance goal achievement. This encompasses proposing achievable intermediate goals—for instance, saving 10% of one's income each month towards a future purchase—and offering a clear timeline to estimate the journey toward the ultimate goal. Additionally, SpendSense identifies potential pitfalls by flagging exceeded limits, and providing timely alerts such as, "Warning: You've surpassed your restaurant spending limit; consider dining at home this week!" The Discover page amplifies this experience by furnishing valuable links to articles and enticing website deals, fostering access to additional information and potential savings.

## II. Project Schedule and Task Planning

Throughout this semester, we've navigated the intricacies of our project through a harmonious blend of both virtual and in-person group meetings held weekly. The visual representation (see Figure 1 in the appendix) is our journey encapsulated in the Gantt Chart we made at the beginning of the semester, a dynamic visualization of our ongoing progress.

Our expedition commenced with a meticulous exploration of project requirements and the envisioning of desirable app features. This laid the foundation for a strategic plan, delineating weekly objectives that became our guiding beacons. To breathe life into our collective vision, we translated concepts into tangible forms by crafting intricate hand-drawn models, providing a vivid preview of the application's aesthetics and functionality.

Our collaborative efforts seamlessly converged within a dedicated team GitHub repository, acting as a collaborative canvas where individual strides and collective progress were vividly showcased. The programming orchestration unfolded with finesse, harnessing the power of

Visual Studio, React Native, Figma, and Expo Go to bring our vision to life. Through this dynamic amalgamation of tools and technologies, our project unfolded as a testament to both creativity and technical prowess.

The requirements phase of our project spanned from the week of August 20 through the week of September 10th. During this time, our focus was to define the requirements and goals for our project, including any pages, features, and design capabilities to enhance user experience. We developed a project plan including a list of deliverables, milestones, meeting dates and times, and preferred communication methods. We scheduled weekly meetings on Tuesdays at 2:00 pm at the Lawrence V. Johnson Library. In the event that any team members were not able to attend a meeting, or we decided to have an additional weekly meeting, we also used Discord to collaborate virtually. Most communication between meetings occurred via SMS group chat.

As we transitioned to the design phase of our project, we began designing a database to hold user account information such as their username, password, and transaction history. We also designed mockups of the *login*, *create account*, and *dashboard* application pages using Figma, as well as a PowerPoint presentation of our ReactNative prototype to present to the class. We also began to rework requirements as needed to fit our goals and expectations for the finished product. This allowed for a more seamless and intuitive application design. This phase spanned from the week of September 19th through the week of October 8th.

Once the design phase had been completed, we moved straight into development. We began to implement our reworked requirements using ReactNative, VisualStudio, and ExpoGo. We used Github as a repository to update our version history and document any changes made. Testing was a collaborative effort, involving group discussions to decide on necessary updates and solutions to any roadblocks encountered. From there, we were able to transition into the monitoring and debugging phase. We completed the design for our website (see section V) and are currently working on the last stage of application debugging.

### III. Software Development Components

*ApplicationService* - Located in the core directory of the application, ApplicationService handles user authentication and data persistence by saving the state of an active user (username and password). It controls the data presented on the application depending on the user and allows for changes to be made dynamically for debugging purposes. See Figure 7 in the appendix to view this source code.

*User* - Accepts the user and password as its parameters. ApplicationService creates an instance of a User and sets the activeUser variable to the most recent login. Upon creation, the timestamp of when the user registered is saved along with a generated key in the local database. This ensures the same user does not register more than once. The User is our single source of truth since it stores all the required data needed for the application to function. A User is successfully created when the passwords match.

*DatabaseService* - New User logins are stored in a Map variable and saved with a UUIDv4 key to ensure there are no duplicate sessions. The addUser function searches through all users and validates new users during account creation. Users that have already created an account return false. DatabaseService provides access to all users which is needed for user-to-user interactions in our application.

*State Management* controls global access for userData and is managed by UserContext. React comes with the “useState” method that tracks changes for events like “onPressed” and allows us to call functions with buttons and text fields. For the account creation screen, useState was used for the “onChangeText” property of the username and password text field. As a result, the username and password variables get updated whenever a user adds or removes text from each respective text field.

## IV. Mobile App

During the brainstorming phase, Elliot and Reginald took on the pivotal task of laying the foundation for the backend development. Their focus revolved around creating robust skeletons that would form the backbone of our project. Simultaneously, Jade and Tori delved into the side of design and documentation, starting with hand-drawn notes and models to determine which types of graphs would work best for our project (see Figure 2 in the appendix). These initial sketches evolved into mockups (figure 3) of the *welcome*, *discover*, and *dashboard* pages.

Each graph adds to the conceptualization of the app’s visuals and user interface experience. We ultimately decided on a line chart tracking a user’s balance over time, which includes a shaded area representing a “danger zone” if they drop below the minimum balance required to meet their predefined goals. We also decided to use a pie chart to categorize the user’s spending into spending types, such as *bills*, *groceries*, *subscriptions*, *tuition*, and *misc*.

With a clear vision of functionalities and layout, we seamlessly transitioned into the coding phase. Our canvas for mobile application pages took shape within Expo Go, while the programming realm was masterfully navigated using Visual Studio, powered by the dynamic trio of JavaScript, HTML, and CSS. This collaborative effort marks the fusion of creativity and technical prowess, a symphony of skills that brought our envisioned project to life. See Figure 4

in the appendix for images of the *welcome*, *create account*, *login*, *dashboard*, and *discover* pages. Figure 5 displays the *discover* page on the web application.

The welcome page redirects users to log in or create an account, depending on whether they have used SpendSense before. The dashboard provides all of the user's information regarding savings, points, balance, and spending charts. This allows users to track their financial situation at a glance. It also allows them to add a transaction that will be added to the statistics and transaction history. The *discover* page contains clickable links to apps and stores to find great deals and coupons, shop second-hand, and locate local flea markets and food banks. Having these links available at a glance will allow for smarter shopping and finding affordable or free resources when finances are tight.

## V. Website

For the website, we initially split our information into a Home, Project Selection, and SRS page. The Home page contained information about each team member as well as their role in the project. The project selection and SRS pages provided general information about the application and its features. Because none of us were very familiar with HTML and CSS prior to this semester, building a website from scratch required a steep learning curve at the beginning. However, we ultimately overcame this roadblock by meticulously learning these languages and refining our code.

After a few weeks of designing and redesigning our website, we ultimately decided to consolidate all of the necessary information onto a single webpage. See Figure 6 in the appendix.

## VI. Version Control

Our version control strategy centers around leveraging the robust capabilities of GitHub to maintain an organized and collaborative development environment. We have established a foundational repository on GitHub to serve as the bedrock for our project. Within this platform, we orchestrate our development workflow by utilizing GitHub's versatile features, specifically the issues and milestones functionalities.

To articulate the project's requirements effectively, we outline them using GitHub's issues feature. Each requirement is encapsulated as an issue/task, providing a clear, granular view of the tasks at hand. This approach not only streamlines our development process but also ensures that every aspect of the project is meticulously documented and easily traceable.

Furthermore, GitHub milestones serve as a roadmap, allowing us to group related issues under a common objective. This not only enhances project organization but also provides a visual representation of our progress as we move toward achieving specific milestones.

## VII. Challenges, Assumptions, and Risk Assessments

The development of SpendSense presented challenges in balancing backend intricacies with design efforts, demanding continuous collaboration. Most of our team members lacked experience using ReactNative, HTML, and CSS prior to this semester. Mastering these skills involved a steep learning curve that required several hours of learning, trial, and error before they could be applied to the project. Assumptions centered around user engagement and data accuracy, influencing data model design and security measures. Risks, such as reliance on third-party APIs, were mitigated with contingency plans, ensuring project resilience. The internationalization challenge prompted a flexible approach to language and currency, reinforcing the importance of adaptability and strategic planning in navigating project complexities.

## VIII. Conclusion

The journey of creating SpendSense has been marked by collaborative dedication, strategic planning, and creative innovation. From the inception phase, where Elliot and Reginald laid the backend foundation, to the design and documentation efforts led by Jade and Tori, every step was a testament to our collective commitment.

GitHub served as the central hub for version control, encapsulating our progress and contributions. Weekly meetings provided a dynamic space for ideation and planning, reflected in the detailed Gantt Chart tracking our ongoing journey.

Design considerations, including the choice of a line chart and pie chart, showcased a balance between aesthetics and user-centric functionality. Our programming endeavors, utilizing Visual Studio, React Native, Figma, and Expo Go, translated conceptual features into tangible functionalities.

The logical data model and data dictionary underscored our commitment to data integrity and security. Usability, performance, and security were prioritized, ensuring SpendSense delivers a seamless and secure financial management experience.

In conclusion, SpendSense embodies our collective passion for technology and financial empowerment. Each team member's contribution, from brainstorming to the final lines of code, has resulted in an innovative solution to simplify personal budgeting.

# Appendix

Figure 1

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUUE DATE	PCT OF TASK COMPLETE
<b>1</b>	<b>Requirements Phase</b>				
1.1	Project Selection	Team	8/20/23	8/27/23	100%
1.2	Define Requirements	Team	9/12/23	9/16/23	100%
1.3	Project Plan	Team	9/3/23	9/10/23	100%
1.3.1	Meet with Professor Perry	Tori	9/12/23	9/12/23	100%
<b>2</b>	<b>Design Phase</b>				
2.1	Define Working Tools	Team	8/20/23	8/27/23	100%
2.2	Design Database	Tori	9/19/23	9/26/23	100%
2.3	Design Mockups	Jade	9/21/23	9/28/23	100%
2.4	Develop Working Prototype	Elliot	10/3/23	10/10/23	100%
<b>3</b>	<b>Development Phase</b>				
3.1	Review Prototype Design		10/3/23	10/8/23	100%
3.2	Complete React Native Tutorials	Team	9/12/23	9/23/23	100%
3.3	Rework Requirements	Team	10/3/23	10/8/23	100%
3.4	Implement Requirement Updates	Reginald	10/17/23	10/26/23	90%
3.5	Document Updated Design	Tori	10/24/23	11/2/23	90%
3.6	Test Application	Jade	10/24/23	11/2/23	90%
<b>4</b>	<b>Project Performance / Monitoring</b>				
4.1	Presentation Preparation	Team	11/1/23	11/7/23	100%
4.2	Create Poster/PowerPoint	Team	11/1/23	11/7/23	100%
4.3	Submit Final Report to D2L	Tori	11/1/23	11/7/23	0%

Figure 2

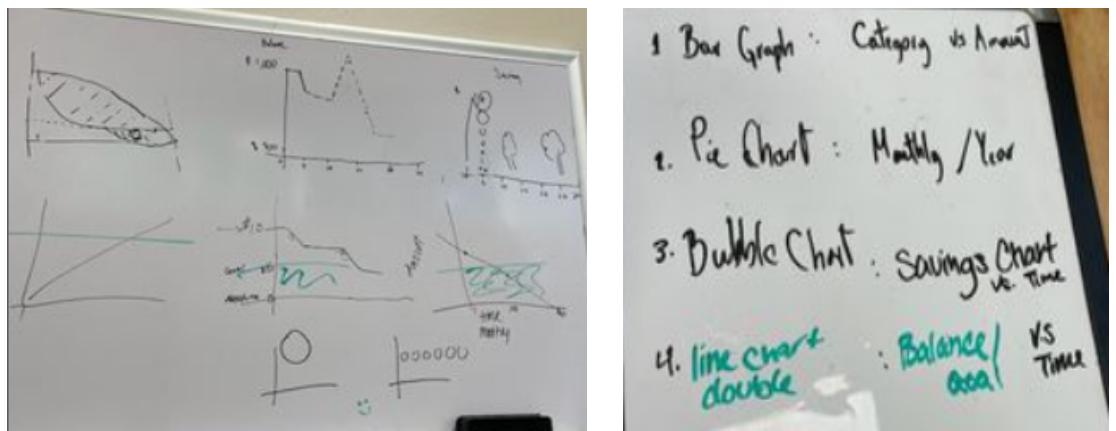


Figure 3

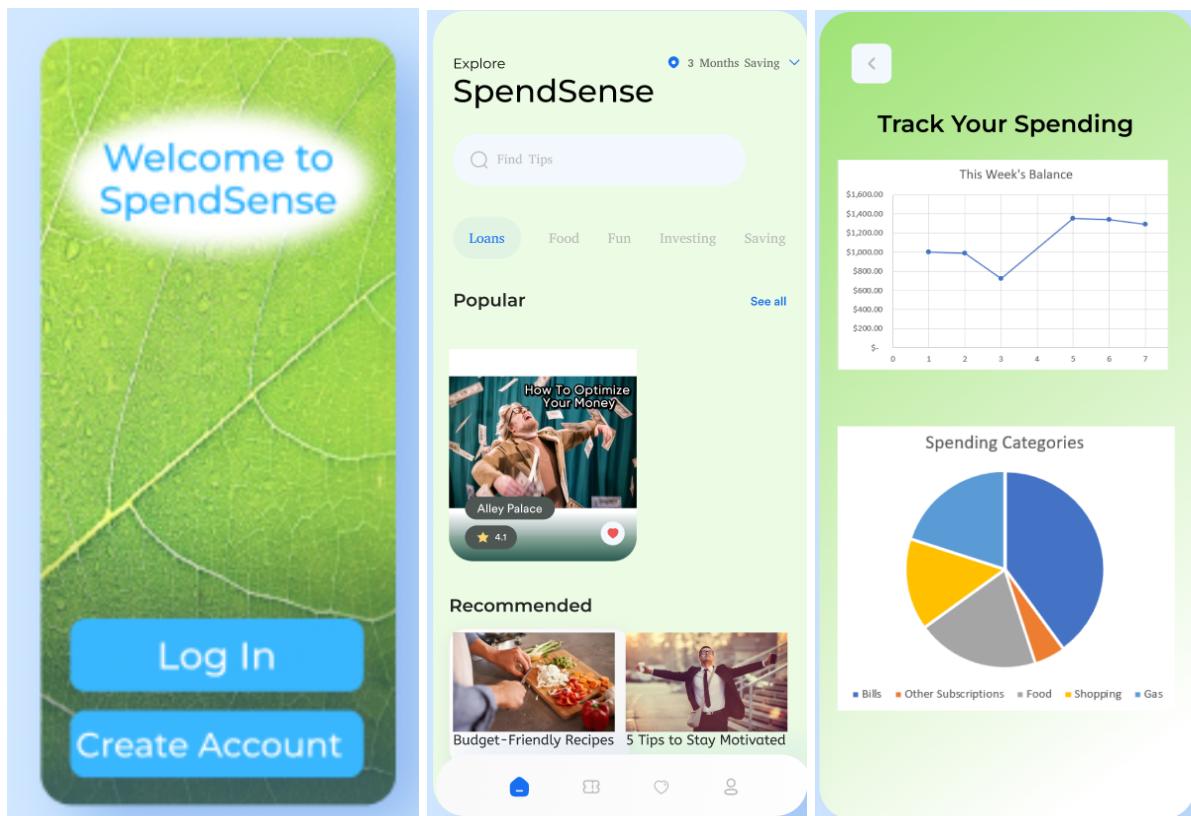


Figure 4

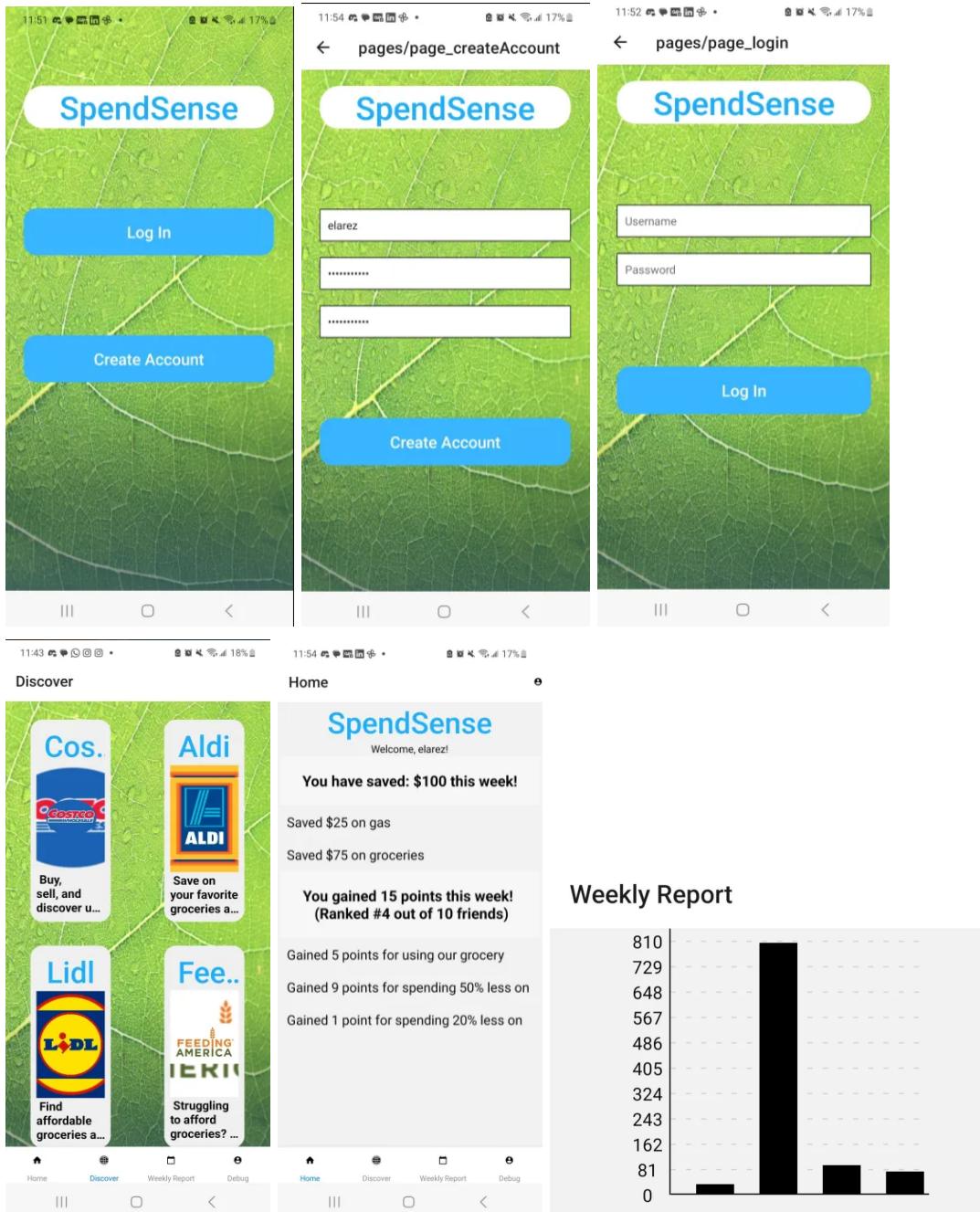


Figure 5

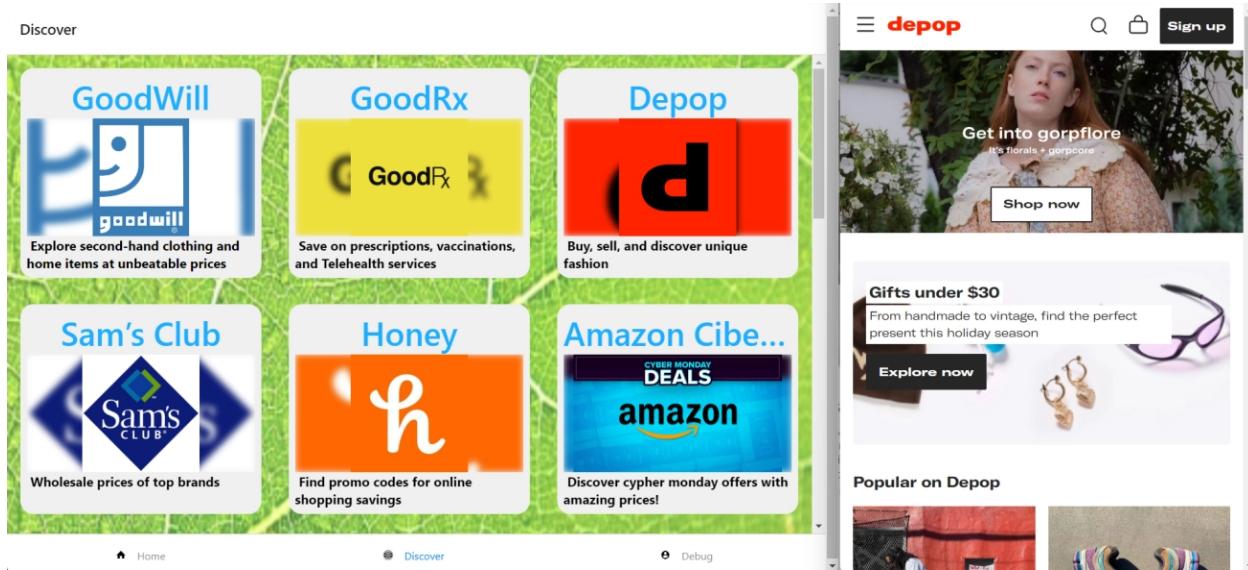


Figure 6

The landing page has a large green leaf background. At the top center is the title **SpendSense** and below it, the subtitle **Your Personal Finance Companion**.

The page is divided into four main sections:

- SP-25 Blue**: Features four team member portraits (two men and two women).
- We are SP-25 Blue**: Describes the Senior Project Team 25 (Blue) as a group of computer science seniors working on mobile development. It mentions they are using React Native, HTML, and CSS to achieve learning objectives in CS 4850.
- Welcome to SpendSense**: Provides an overview of the app, stating it is a personal budgeting app for young adults navigating financial responsibilities in college. It tracks balance, spending trends, and progress towards goals.
- Why SpendSense?**: Explains the app's purpose: providing user control over finances through tracking, budgeting, goal planning, and detailed insights.

**Key Features** listed include:

- Effortless Expense Tracking through intuitive graphs
- Smart Budgeting Tools to predict and anticipate financial necessities
- Intuitive Financial Goal Planning
- Detailed Financial Insights in a weekly basis
- Exploration of saving opportunities, through smart advertising (coupons/offers)

A call-to-action button **Get Started** is located at the bottom right.

Figure 7

```
MobileApp > SpendSense > app > (tabs) > TS page_authentication.tsx > ...  
18  
19  export default function PageAuthenticationScreen() {  
20    return (  
21      <ImageBackground source={require('../assets/images/Backgrounds/Leaf.png')} style={styles.image}  
22        <View style={styles.container}>  
23          <Text style={styles.title}>  
24            Welcome to SpendSense  
25          </Text>  
26          <Pressable onPress={() => {  
27            const myPayment = new Payment(100, "Initial")  
28            myPayment.print()  
29            ApplicationService.changeUser('Elliot Larez', "123")  
30            ApplicationService.printUser()  
31            ApplicationService.changeUser('Reginald Appiah', "123")  
32            ApplicationService.printUser()  
33          }}>  
34            style={({ pressed }) => [styles.button1, { opacity: pressed ? 0.5 : 1 }]}>  
35              <Text style={styles.button1_text}>  
36                Log In  
37              </Text>  
38            </Pressable>  
39            <Pressable style={({ pressed }) => [styles.button1, { opacity: pressed ? 0.5 : 1 }]}>  
40              <Text style={styles.button1_text}>  
41                Create Account
```