University of Regina | Go far, together.

ENSE 375 – Software Testing and Validation

DollarDiary

- **Team Members:**

1. **Aliasgar Saifeebhai Mithawala – 200517808**
2. **Savan Rameshbhai Pedhadiya – 200517532**
3. **Jay Bharatkumar Panchal - 200518664**

3. Solution:

1) Solution - 1:

The first solution involves developing the DollarDiary application using Java with the Spring Boot framework with basic CRUD (Create, Read, Update, Delete) functionality for expenses.

- **Reason for not selection this solution:**

  1) **Limited Functionality:** This solution only provides basic expense tracking without additional features such as categorization or budgeting, which limits its usefulness.
  2) **User Engagement:** Users might not find the application useful for long-term financial planning without the ability to categorize expenses or set budgets.
  3) **Market Competitiveness:** The lack of advanced features means this solution would not stand out in a market where many applications offer more comprehensive financial management tools.

2) Solution - 2:

The second solution builds on the first by incorporating expense categorization and budget setting.

- **Advantages over solution - 1:**

  1) **Enhanced Functionality:** Including categorization and budget setting makes the application more useful for users trying to understand and manage their spending patterns.
  2) **Better User Engagement:** These additional features help users track their finances more effectively, leading to better retention.
  3) **Partial Implementation of User Feedback:** This solution addresses some of the key features requested by users, making it more attractive where many applications offer more comprehensive financial management tools.

- **Reason for not selection this solution:**

  1) **Scalability Issues:** Adding categorization and budget setting without considering scalability might lead to performance issues as the number of users grows.
  2) **User Experience Complexity:** While adding these features improves functionality, it may also complicate the user interface, making it less intuitive for users who prefer simplicity.

3) **Integration Challenges:** Implementing these additional features might introduce complexities in integrating and maintaining the application, leading to potential technical debt and increased maintenance efforts.