Project Completion Review (PCR) Report

Project Overview

- **Project Name:** BillSyncEase
- **Project Duration:** January 11, 2024 to April 1, 2024
- **Project Team:**
    - Aadee, Hannah, Shubhdeep
- **Product Owner:** Sarah
- **Scrum Master:** Gurkaran

Executive Summary

BillSyncEase is a medical billing application that makes it easy for patients and healthcare providers to facilitate communication about insurance claims and receipts. Some insured services, such as orthodontics, dental care, and eye care cannot be directly billed to insurance companies; patients must submit their own claims to get reimbursed for the amount they paid. People unfamiliar with the process may find this stressful and may even avoid using the health insurance they have through their jobs. The main functions of the medical billing system will be found all in the app, eliminating the need for a bunch of different apps, and encouraging patients to utilize their benefits.

Key achievements:

- Created a functional prototype at the end of each sprint
- Made a user-friendly platform

Challenges:

- Time constraints due to conflicting schedules

Project Objectives

Our main objective is to create a medical billing system application that features user registration, and medical records tracking, while ensuring it is user-friendly and has usable functions at the end of each sprint.
Our goal is to help users make insurance claims, manage and track their medical bills, and payment processing to make the billing process easier for both patients and healthcare providers.
Initial objectives and goals:

- Account creation and user sign-in
- Home page
- Claim submission
- Account management

Adjustments to project:

- Adjusted user interface based on feedback from users, professor, and team members

Sprint Overview

Sprint 1

- **Duration:** January 11, 2024 to January 25, 2024
- **Key Deliverables:**
    - Wireframe
    - Mockup
    - Product Backlog
- **Achievements:**
    - Proper Mockup
    - Basic Product Backlog
- **Challenges:**
    - The biggest challenge we experienced was all agreeing on a base. Everyone had their own idea of what the app should have or look, eventually we all managed to compromise on a basic design.

Sprint 2

- **Duration:** January 25, 2024 to February 8, 2024
- **Key Deliverables:**
    - Basic design Prototype
    - System architecture
    - Design ERD
    - User flow Diagram
- **Achievements:**
    - Basic design Prototype
    - Updated and reworked Product Backlog
- **Challenges:**
    - During this time our main challenge was understanding how figma worked and getting used to it and good enough at it to use it effectively.

Sprint 3

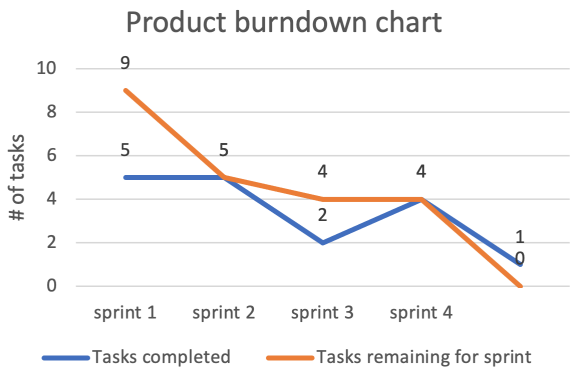
- **Duration:** February 8, 2024 to March 6, 2024
- **Key Deliverables:**
    - Usable prototype.
- **Achievements:**
    - Function to make a claim
    - Function to view previous Claims
    - Notification functions
- **Challenges:**
    - The biggest challenge we faced at this point was “how are we going to implement this?” there were components that we wanted to implement however we were having trouble visualizing.

Sprint 4

- **Duration:** March 6, 2024 to April 1, 2024
- **Key Deliverables:**
  - Functioning Prototype
- **Achievements:**
  - Implemented EHR system
  - Added a help page
- **Challenges:**
  - During this Sprint the only challenge we faced was making sure we were not rushing to complete the project on time.

Project Metrics

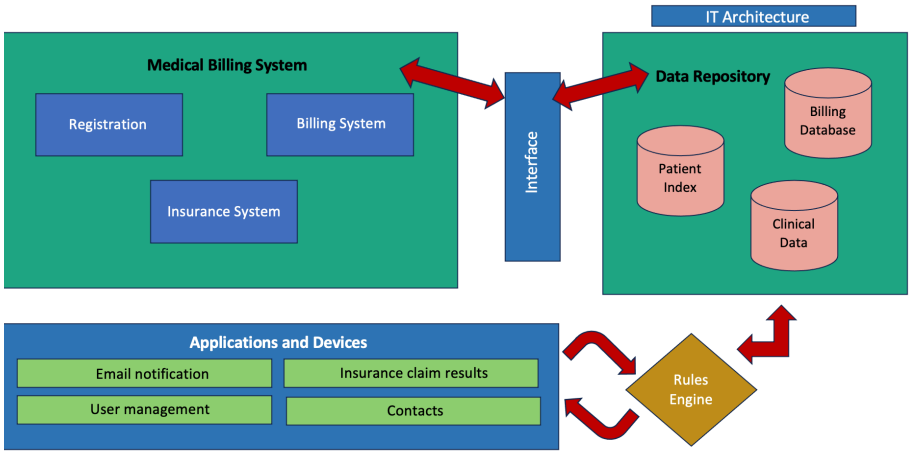
We evaluated our progress on the project by tracking our tasks using an editable product backlog. The scrum master classified each task as To be started, Work in progress, or Done. The product burndown chart is based on the product backlog and it shows how many tasks were remaining at the beginning of each sprint and how many tasks were completed at the end of each sprint. For example, we had nine tasks that we wanted to finish by the end of sprint 1. At the end of the sprint, we actually completed 5 tasks; the rest (4) were in progress. We started a bit slow, but ended strong.



Product Backlog Evolution

The Product Backlog for our project had more or less stayed the same. The changes that had occurred regarding the product backlog was how each of our items/functionalists would look or be implemented. For example, one of the things on it was integrating an EHR system which we did by adding the ability to see the doctor's note on a claim for a certain medication. This is just one of the parts that we did this too; there are a few others as well.

System Architecture



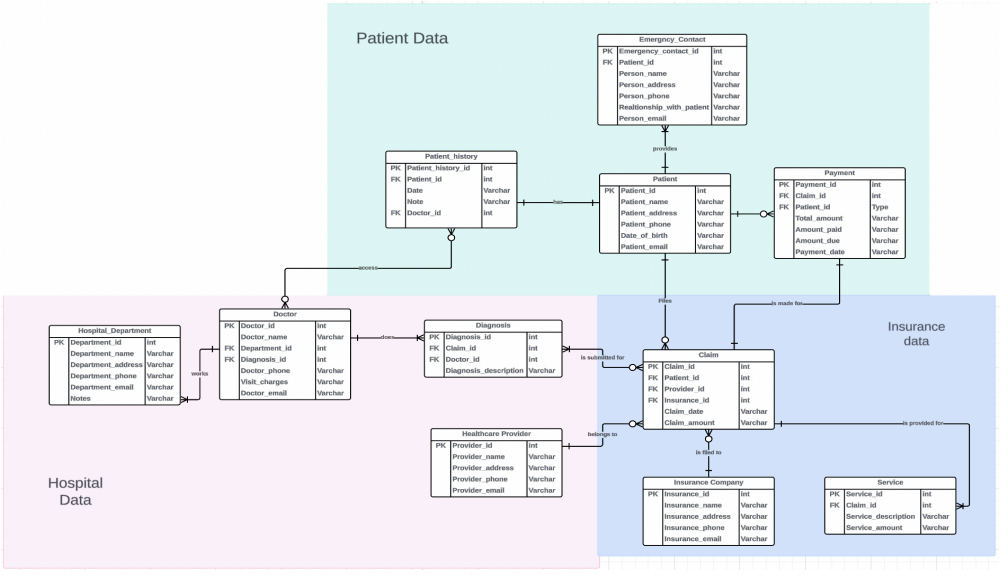
Data Repository: Large database that is used to manage and store data for analysis

Clinical data = Medical history
Patient index = ID, contacts
Billing database = Addresses, receipts

Applications and devices: Functions that will be in the app

Rules Engine: Deals with authorization and rules

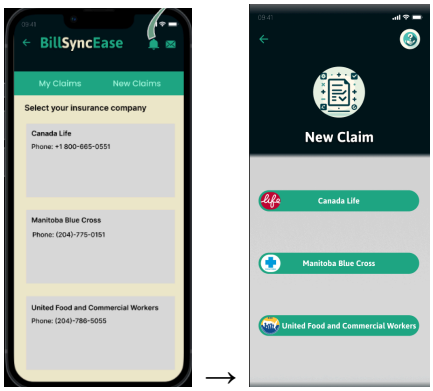
Database Design



User Interface Design

- Large and easy to read font size so it is accessible for everyone, especially people who have reading difficulties
- Complementary colours with good contrast
- Few colors so that it is not too distracting and users are not overwhelmed
- Included more graphics, such as insurance company logos to make them easier to identify

Before / After of Selecting Insurance Page



Testing Strategies

1) **Usability Testing**: Team members can assess the ease of use and navigation of the prototype by performing tasks typical of end-users. They can provide feedback on the intuitiveness of the interface, clarity of instructions, and overall user experience.
2) **Functionality Testing**: Team members can test the functionality of different features of the prototype to ensure they perform as expected. This involves verifying that all buttons, links, and interactive elements work correctly and that data input and retrieval functions properly.
3) **Performance Testing**: Team members can evaluate the performance of the prototype by testing its loading speed, response time, and overall responsiveness. They can simulate different usage scenarios to assess how the prototype handles various levels of user interaction and data processing.
4) **Accessibility Testing**: Team members can test the prototype for accessibility compliance to ensure it is usable by individuals with disabilities. They can assess factors such as keyboard navigation, screen reader compatibility, and color contrast to ensure the prototype meets accessibility standards.

Release Plan

1. Initial Release (Version 1.0):
    - Features:
        - User registration and login functionality.
        - Profile management features, including notifications, emails, and editing profiles.
    - Rollout Strategy:
        - Conduct a soft launch of the application to a limited user base for initial testing and feedback.
        - Gather user feedback through analytics, user reviews, and usability testing sessions.
    - Considerations:
        - Ensure the application design is adaptable and responsive, catering to different devices and platforms.
        - Provide user guidance and tooltips within the application interface to assist users in navigating profile management features effectively.
2. First Major Update (Version 1.1):
    - Features:
        - Submission of insurance claims (PCR functionality) optimized for intuitive use.
    - Rollout Strategy:
        - Release the updated application to all registered users after thorough testing and validation.
        - Notify users about the new functionality through in-app notifications, email updates, and release notes.
    - Considerations:
        - Offer interactive walkthroughs or tooltips within the application interface to educate users on how to submit insurance claims effectively.
        - Monitor application performance metrics and user feedback closely after the update to identify and address any usability issues.
3. Subsequent Updates (Version 1.2 and beyond):
    - Features:
        - Enhancements to PCR functionalities, including additional features such as requesting assistance or support directly from the application interface.
        - Addition of new features based on continuous user feedback and market demand.
    - Rollout Strategy:
        - Implement incremental updates rolled out periodically, aligned with development cycles and user priorities.

        - Prioritize feature enhancements based on continuous user feedback and Agile backlog refinement sessions.
    - Considerations:
        - Engage users through surveys or feedback forms to gather insights for future enhancements and improvements.
        - Conduct thorough testing on new features to ensure they align with user expectations and enhance the overall user experience.
4. Future Versions (Version 2.0 and beyond):
    - Features:
        - Expansion of application features and functionalities, including support for additional platforms, advanced analytics, and integration with third-party services.
    - Rollout Strategy:
        - Plan for major version releases with significant feature enhancements or architectural changes, following an Agile release planning process.
        - Communicate changes and improvements through release notes, in-app announcements, and email notifications.
    - Considerations:
        - Ensure consistent user experience and functionality across different devices and platforms.
        - Continuously monitor application performance and user engagement metrics to iterate on features and prioritize future development efforts.

Documentation

Summary of the documentation of project for BillSyncEase:

1) **User Manual**:

- The user manual provides detailed instructions on how to use the BillSyncEase app effectively.
- It includes step-by-step guides for various user actions, such as signing up, logging in, creating a profile, submitting insurance claims, and managing notifications.
- Each section of the manual is accompanied by screenshots or illustrations to visually guide users through the process.
- Troubleshooting tips and FAQs are included to address common user queries or issues.

2) **Developer Documentation**:

- Developer documentation outlines the technical aspects of the BillSyncEase app for future reference and maintenance.
- It includes information on the app architecture, data models, APIs, libraries, and frameworks used in development.

- Detailed instructions are provided for setting up the development environment, running tests, and deploying updates.
- Code snippets, diagrams, and diagrams may be included to aid developers in understanding and modifying the app codebase.

3) **How-to Guides**:
- How-to guides offer concise instructions on specific tasks or features within the app.
- Examples include guides on signing up, logging in, creating a profile, viewing bills and insurance claims, submitting a claim, and managing notifications.
- These guides are designed to be user-friendly and accessible, providing clear and actionable steps for users to follow.

Lessons Learned

As a group, we definitely learned a lot about project management. The Agile process was new to some of our group members, so it was a struggle at first to adapt to an incremental, iterative development process as opposed to a sequential one. We worked well as a team in dividing work to highlight each individual member's strengths, whether that was Figma design, product backlog management, or report writing. As a team, we made sure to quality-check each member's work to ensure that we all felt comfortable with what we were presenting.

In terms of improvement, our sprint management leaves room to be desired. This includes making sure that everyone is on the same page, as sometimes it was unclear which specific features or deliverables had to be completed at what time. However, our team was able to overcome these challenges through communication, and asking clarifying questions when we were unsure of the intention. We believe that this project has been beneficial to our adherence to Agile principles.

Stakeholder Feedback

Throughout the course of this project, the development of our product was aided by important stakeholder feedback. When presenting our project proposal, the main feedback we received was about the structure of our product backlog. We were advised to incorporate user stories, which outline the purpose of each proposed feature. This would also help our project align more with the core values of Agile, specifically with user-centered design. As such, we updated our product backlog to be much more exhaustive, including said user stories as well as more detail about the type of user and the status of feature implementation. For our mid-project report, the feedback we received was largely centered on making our product more user-friendly. This included clarifying our user flow diagrams, using large icons to direct users to specific services, and choosing colours with high contrast.

Agile Methodology Assessment

The Agile methodology was very effective for this project. The concept of iterative design allowed us to go back and edit our previous work based on feedback, which resulted in a better final product than if we had used a sequential, waterfall approach. The methodology facilitates lots of communication between not only team members, but stakeholders and others interested in the project as well. This helped us brainstorm different ideas of features and their implementation, so that our project contained several unique, defining features and was distinct from other medical billing platforms on the market.

Recommendations

For future Agile projects, we would suggest setting a consistent meeting time. For example, holding all of your team meetings on Sundays at 4:00pm. This way, the schedule and expectation is set from the start, so group members can plan around the meeting time. We would also suggest more frequently looking back and reflecting what was done previously. This clearly includes reevaluating prior work and making changes, but should also include reflecting on past meetings, and evaluating their effectiveness. That way, we can see what improvements we can make to our methodology in addition to the project itself.

Conclusion

In conclusion, we believe that this project has been a great success for our team. As an introductory method for some and a seasoned method for others in our group, many facets of Agile were learned and improved upon throughout the semester. We learned the importance of user-centered design, as well as the importance of facilitating communication between group members, stakeholders, and anyone else interested in our work. We would like to thank Professor Balogun for his valuable feedback and evaluation of our project, and for giving us an opportunity to get hands-on experience. Overall, we as a group are proud of what we have achieved this semester, and look forward to more opportunities like this one.
