Process Deliverable

Jordan Atanassov, Reagan Gary, Omer Abbas, Jake Lehoullier, Sebastain Bukvic

- System Requirements
    - Application should be whitelisted on the company network
    - User Friendly Interface for submitting feedback and generating reports
    - Applications should be encrypted in the database in case of a security threat
    - Application should work alongside existing project management tools
    - Scalable to support traffic from multiple teams at the same time
- User Requirements
    - Users should be able to flag tasks or instructions they find unclear
    - Team members should provide additional comments or clarification
    - Team leader should be able to review the report
    - Team leader should be able to take appropriate action
- Non-Functional Requirements
    - The application should have a responsive design that is adaptable to different devices
    - Reliable up time
    - Maintainable and extensible, allowing for easy addition of features
- System Architecture
    - Client Side
        - Application will utilize React-JS and CSS for the front end
        - The application will utilize NodeJS to do calls to the server
    - Server-side
        - The server will run on NodeJS to handle data processincxg and storage
        - The server will also handle user authentication, access control, and data encryption
    - Database
        - The application will make calls to a database written in MySQL which will provide easy storage, and access of a large amount of data
- Project Timeline and Milestones
    1. Phase 1, Planning and Design
        i. Gather requirements for stakeholders
        ii. Design use cases
        iii. Wireframe and make diagrams and schemes of the architecture
    2. Phase 2, Development
        i. Set up development environment
        ii. Implement the client-side section of the interface
        iii. Develop server side Api endpoints
        iv. Implement user authentication
        v. Develop the database and encryption rules
        vi. Implement feedback reporting and email routing
    3. Phase 3, Testing and Deployment
        i. Unit Testing

    ii. Perform user acceptance testing
    iii. Address bugs found in development and testing
    iv. Deploy application to a testing environment
    v. Deploy to the production environment
4. Phase 4, Training and Rollout
    i. Conduct training sessions for team members and leaders
    ii. Gather feedback
    iii. Monitor server performance
5. Phase 5, Maintenance
    i. Provide support
    ii. Additional tools upon request
    iii. Security patches when needed