Client Handover Document

Implementation and Handover Documentation

Project Details

- **Project Name:** OnTime App
- **Client:** Vaughan Randell
- **Organisation:** PacBuild
- **Project Team:** OldTimers
- **Date of Handover:** 07 Jun 2024

Product Description

The OnTime app is a timesheet management system designed for the PacBuild organisation. It allows employees to create timesheets, including logged work hours, supervisors to approve or disapprove timesheets, and admins to manage all app features, including user and job site management. The app uses React Native for the front end and a REST API for the back end, with data stored in a MongoDB database.

Product Deliverables

As outlined in the project proposal, the product deliverables include:

- **Authentication and Authorisation:** Secure login and role-based access control.
- **Timesheet Management:** Functionality for employees to log work hours and submit timesheets.
- **Supervisor Access:** Features for supervisors to approve or reject submitted timesheets.
- **Admin Access:** Comprehensive administrative controls for user and jobsite management.
- **Timesheet Exporting:** Ability to export timesheet data to CSV for reporting purposes.
- **UI Interface:** User-friendly interface for seamless navigation and usability.

Implementation Details

Development Environment Requirements:

Frontend

- **Operating System**
  - macOS 11.3 or later is required to run XCode 13 and deploy the app to the App Store. A lower version of macOS may be suitable for development, but 11.3 or later is highly recommended for development and deployment.
  - Windows 7 or later for Android development
- **Software**
  - Your preferred Code Editor: Visual Studio Code is highly recommended
  - XCode 13: Necessary for building and deploying the application to the App Store
  - Node.js and npm: Essential for managing packages and dependencies
- **Dependencies**
  - Project dependencies are listed in the 'package.json' file, which can be found in the project repository on GitHub: OnTime Repository

Backend API Integration:

- **REST API:** The application integrates with a RESTful API
- **Database:** MongoDB is used as the database for the application
- **API Interaction Tools:** Postman is used for testing and interacting with the REST API
- **Endpoints:** Specific API endpoints and their usage details are documented within the project database

Backend

- **Operating System**
  - Windows 10 and later or Mac OS 11.0 or greater
- **Software**
  - Your preferred Code Editor: Visual Studio Code is highly recommended
  - Optional: Postman - for testing endpoints
- **Dependencies**
  - Project dependencies are listed in the 'package.json' file, which can be found in the project repository https://github.com/hannahgmacca/ontime-express-api

Deployment Procedures:

Frontend

- **Deployment to Apple App Store** (using App Store Connect and XCode)
    1. Open XCode
    2. Navigate to Product Scheme Edit Scheme
    3. Change the build configuration to 'Release' and close the window
    4. In the second drawer on the left, go to Targets OnTime General and increment the build number by 1 (and update the version number if needed)
    5. Go to Product Archive and follow the prompts to create the archive
    6. After archiving, the application can be uploaded to the App Store Connect for distribution
- **Post-deployment Cleanup**
    1. Revert the 'OnTime.xcscheme' to set debug as the current scheme
    2. Commit any build or version changes to the Git repository

Backend

```
# installs nvm (Node Version Manager)
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.7/install.sh | bash
# download and install Node.js
nvm install 20
# verifies the right Node.js version is in the environment
node -v # should print `v20.14.0`
# verifies the right NPM version is in the environment
npm -v # should print `10.7.0`

# clone backend repository
git clone git@github.com:hannahgmacca/ontime-express-api.git

# navigate to application folder
cd ontime-express-api

# install dependencies
npm install

# install vercel CLI tools (global)
npm i -g vercel

## MUST BE LOGGED INTO VERCEL IN BROWSER BEFORE NEXT STEPS

# run vercel deployment
vercel --prod ontime-express
```

Hosting Information

The application is hosted on ... Below are the details for accessing the hosting environment:

Frontend

- Hosting Provider: LYMAH DATA
- Server: App Store
- Admin Access: (Provided to the client for management and maintenance purposes.)

Backend

- Hosting Provider: Vercel
- Server: Serverless Express API
- Admin Access: (Provided to the client for management and maintenance purposes.)

GitHub Information

To access and modify the application files, the following GitHubs can be used:

Frontend Application: https://github.com/BobDeUncle/OnTime/tree/main/OnTime

Backend API: https://github.com/hannahgmacca/ontime-express-api

Local Testing

Backend

API Documentation: https://documenter.getpostman.com/view/16152243/2sA3BheZmw

Download Source Code: Obtain the source code and database from the provided links:

- **Source Code:** https://github.com/hannahgmacca/ontime-express-api
- **Environment Variables:**

```
CONNECTION_STRING = 'mongodb+srv://admin:MUwtTPbSnL6SIGTg@ontime.hdjh4sr.mongodb.net/?
retryWrites=true&w=majority&appName=OnTime'
SECRET_KEY = 'ontime-secret-key'
NODE_ENV = 'development'
RECOVERY_EMAIL = 'ontime.recovery.emails@gmail.com'
RECOVERY_PASSWORD = 'ubsz wgig skfi vudg'
TEST_EMAIL = 'email-testing-239aac4a-c997-47f7-b8ee-5f96c695d9df@email.devit.software'
```

Node JS Environment set up

```
# installs nvm (Node Version Manager)
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.7/install.sh | bash
# download and install Node.js
nvm install 20
# verifies the right Node.js version is in the environment
node -v # should print `v20.14.0`
# verifies the right NPM version is in the environment
npm -v # should print `10.7.0`

# clone backend repository
git clone git@github.com:hannahgmacca/ontime-express-api.git

# navigate to application folder
cd ontime-express-api

# install dependencies
npm install
```

- Example for setting up MongoDB: Non-required as staging and production databases are cloud instances.

2. **Run Tests:**
   - Ensure all functionalities are working as expected in the local environment.

Remaining Tasks

List of tasks that are pending completion:

- Final Testing: (Ensure thorough testing of all features before full deployment.)
- Bug Fixes: (Address any bugs or issues identified during testing.)
- Documentation: (Finalize and deliver any additional documentation required by the client.)

Acceptance of Product by Client

Purpose of this Document & Intellectual Property Agreement:

This document serves to officially hand over the OnTime app to the client, Vaughan Randell, on behalf of PacBuild. All intellectual property related to the OnTime app is transferred to the client upon acceptance.

Team Signatures

- **Student 1 Full Name:** Jesse Mansbridge
  - Signature: image.jpg
  - Date: 7/06/2024

- **Student 2 Full Name:** Hannah McDonald
  - Signature: Hannah McDonald
  - Date: 6/06/2024

- **Student 3 Full Name:** Jessica Pacini
  - Signature: Jessica Pacini
  - Date: 5/06/2024

- **Student 4 Full Name:** Elijah Yeaman
  - Signature: eyeaman
  - Date: 07/06/2024

Client Signature

- **Full Name:** Vaughan Randell
  - Signature: image-2024-5-24_12-40-52.png
  - Date: 7/06/2024