Project Status Report Team1

<Book Exchange>

Submitted: < Mar 5, 2020>

|  |  |
| --- | --- |
| **Prepared By:** | Machida Hiroaki |
| **Date Prepared:** | Mar 5, 2020 |

**Accomplishments Since Last Report**

* **Integration test automation framework setup (Hiroaki)**

Now integration test can be run by one command. Cypress was installed into the project and configuration files to run Cypress were created. One integration test case that creates & delete a post for selling book was successfully run.

* **Staging environment setup (Hiroaki)**

Developers and users can access to the latest application via the internet. An AWS EC2 instance was created, necessary packages are installed. A domain, team1.work, was got from a domain registrar, onamae.com. DNS configuration on onamae.com and routes configuration on AWS was done.

* **Continuous integration environment setup (Hiroaki)**

Developers can see the test log, screenshots, and movies via the internet. A shell script on the EC2 instance gets the latest source codes, runs tests, and make test results. Initially, the environment was set up on Linux but failed because “yum” could not install Xvfb successfully nor install Xvfb dependencies manually. Finally, the environment was set up on Ubuntu, and Xvfb and its dependencies were installed manually. Apart from that, a web server, Nginx, was installed, and configurations to separate the routes to the application and directories to the result sets were made.

* **Minimum user story (Hiroaki & Shaurya)**

Users can sell & buy books now. Shaurya initialized the react + redux framework, redux-firestore, and made the database schema design based on user stories and wire-frames. Hiroaki implemented the minimum user story by using Redux as follows. 1. User A registers account 2. User A logs in, 3. User A registers selling book, 4. User B searches for a book, 5. User B finds book 6. User B makes payment, 7. User A sends book and update status, 8. User B receives book and update status.

* **User review for user (Amol)**

User can write a review for another user after receiving books from the user. This is an additional user story to #8 above. One new dialog was created.

After a receiver writes a review, the book status changes to ‘review\_submitted’ and reviewText column contains the text of review.

* **Post for buying book (Shenyuan)**

Users can make a post for buying a book that is not on the market, in contrast to the minimum user story. Most of the implementation was done as a new component.

* **Autocompletion for search (Hiroaki & Peiyan)**

Names of books registered on the application pop on the text field for search. Hiroaki and Peiyan implemented autocompletion by using Material-UI components.

* **Barcode reader (Shenjie)**

ISBN, that is a unique identification for a book, should be automatically input by uploading the bar code of a book. Checked libraries for bar code-reader and each of them didn’t work. The libraries checked are as follows: react-native-camera, webcam, quaggaJS, zbar + WebAssembly.

* **User manual (Hiroaki)**

User manual was created for users to know how to user the application. User manual was created with GitHub flavored markdown and uploaded to GitHub with screenshots, so that everyone can see the manual without downloading.

* **Iteration 1 presentation material (Hiroaki & all)**

The material for iteration 1 presentation is to explain the project to the class. It was created by Hiroaki and checked by all other members.

**Plans to do for Next Report**

* **Integration test (All)**

Integration tests for each implementation should be created by Cypress.

* **Whole user interface improvement (Shaurya + Amol + Hiroaki)**

Whole user interface should be improved. Unnecessary fields, for example, status, that is whether book is clean or not, should be removed. Also, information on transactions should be made easier to see by, for example, making the images of books larger.

* **Advanced search (Shenyuan + Peiyan)**

The application automatically detects categories of books and users can search for books by specifying category. Implement machine learning algorithm to find the categories of books and put it into search result.

* **Barcode reader (Shenjie)**

To find a library that works and move on to implementation.

* **Paypal payment (Shenjie)**

The application should keep the status on whether a user already makes a payment to another user. First, it needs to see if there is any available test environment on paypal.me.

* **Event notification (Hiroaki)**

Notifications for new events should be shown on the dashboard so that users can know what steps they need to take next. A new database schema for notification should be created and the data will be shown to users.

**Key Issues, Risks & Concerns**

|  |  |  |  |
| --- | --- | --- | --- |
| **Item** | **Action/Resolution** | **Responsible** | **Completion Date** |
| [Risk] Dev env setup | Not everyone set up dev env  [Action] Set up dev env in face-to-face  [Result] Everyone has set up the dev env. | Ding Shenjie | 2/25 |
| [Risk] Member skills | Not everyone familiar with React & Redux  **[Action] Follow up questions from members in face-to-face or on Slack** | Sinha Shaurya |  |
| [Risk] Workload outside class | There might be over workload outside class and some members cannot proceed tasks  **[Resolution] Made project rules**  -No-show: agree things without someone not attending unless notified  -Over-due task: change task owner  -Over-due feedback: not reflected in principle | Machida Hiroaki |  |
| [Risk] High workload functions | It’s very likely that there will be conflicts among functions for high workload development  **[Action] In advance, write user stories, screen drafts, and db schemas** | Each member |  |

**Status**

|  |  |  |
| --- | --- | --- |
| **Overall** | **G** | The goal for iteration 1 was achieved. The minimum user story mentioned before has been successfully implemented, and also the staging environment is open on the internet. |

|  |  |  |
| --- | --- | --- |
| Schedule | **G** | The goal for iteration 1 was completed. Not a major issue, but bar code reader, that reads bar code and updates ISBN automatically, was not completed, although it was schedule by iteration 1. |
| Resources | **Y** | Team members not fully utilized. Especially, our team could not utilize well Shaurya, who is good at React and has a role to help others who is new at React, because what help is needed is not communicated well among members. We are arranging more frequent face-to-face meetings. |
| Scope | **G** | In general, the scope is clarified well enough for each iteration.  Iteration 1 implementation done. The minimum functionality, that users can sell & buy books, was implemented as mentioned in the 1st section.  Iteration 2 requirements clarified as mentioned in the 1st section. Detailed screen images need to be defined.  Iteration 3 items are brainstormed and need to be fixed based on the progress of previous iterations and available resources on that period. Details are shown at the end of the schedule part. |

**Schedule**

|  |  |  |  |
| --- | --- | --- | --- |
| **Stage (Sprint)** | **Target Completion** | **Revised Completion** | **Comments** |
| **Iteration1 (Must-have)** | 3/5 | 3/5 | Done |
| Integration test automation framework setup | 2/20 | 2/20 | Hiroaki Done |
| Staging environment setup | 2/20 | 2/20 | Hiroaki Done |
| Continuous integration environment setup | 2/20 | 2/20 | Hiroaki Done |
| Minimum user story | 2/20 | 2/20 | Hiroaki & Shaurya Done |
| User review for user | 3/4 | 3/4 | Amol Done |
| Post for buying book | 3/4 | 3/4 | Shenyuan Done |
| Autocompletion for search | 3/4 | 3/4 | Hiroaki & Peiyan Done |
| User manual | 3/4 | 3/4 | Hiroaki Done |
| Iteration 1 presentation material | 3/4 | 3/4 | Hiroaki & all Done |
| **Iterations2 (Must-have)** | 4/2 |  |  |
| Integration test | 4/1 |  | All |
| Whole user interface improvement | 4/1 |  | Shaurya + Amol + Hiroaki |
| Advanced search | 4/1 |  | Shenyuan + Peiyan |
| Barcode reader | 4/1 |  | Shenjie |
| Paypal payment | 4/1 |  | Shenjie |
| Event notification | 4/1 |  | Hiroaki |
| **Iteration3 (Nice-to-have)** | 4/30 |  |  |
| Book return | 4/29 |  | TBD |
| Auto complete book cover | 4/29 |  | TBD |
| Recommendations engine | 4/29 |  | Amol |

* **Book return (TBD)**

Users can request refunds if there are issues on the book after receiving the book. The user requests refunds on the application and the application should keep track of the status.

* **Auto complete book cover (TBD)**

Application automatically downloads images of book covers from the internet or cache on the application and shows it to users.

* **Recommendations Engine (Amol)**

Recommendations engine is a feature that allows users to receive recommendations of new’ arrival of books based on their interest area. The user dashboard would have a space of recommendations’ from the system that match the user’s interests. User’s interest would be captured through user - preferences.