Sprint 1 Materials

Team 64

[Optional Team Logo]

Zac Ziegler

Will Thompson

Nick Knighton

Tarek Kassim

Cyril John

**[Delete all instructions in red before submitting.]**

# Retrospective Summary

Summarize your retrospective meeting. Discuss what went right and wrong during the sprint, what changes (if any) need to be made regarding procedure, and what adjustments might need to be made to the product backlog. If backlog goals were not met, include an explanation of why not. The summary should be approximately 250 words.

Overall, we felt that our first sprint went well. Although we did not complete everything on the backlog, we were able to make really good progress towards our final product. We were able to create both the manager and server webpages that will be able to make changes to the database in the future. We were also able to connect the front end with the back end so that we can start sending data between to two in order to make the website dynamic. There was a lot that went right, but we also made plenty of mistakes as well; we found that we were not doing a great job of utilizing Jira to properly log work and make sure that everyone is on track. During our retrospective meeting, we talked about how to properly use Jira and how track hours so that everyone was on board with the process. This will make future sprint retrospectives easier because we will be able to better track how much progress was made. We also found that we were not doing a great job of tracking what tickets each person was working on. We had them setup at first, but as time went on and people started completing tickets, they were not assigning new tickets to themselves. This meant that it was difficult to keep track of what was currently being worked on and by who. We also discussed this and decided that we would discuss what tickets everyone is working on at each SCRUM meeting during each sprint. This way we can make sure that everyone knows what is going on and there is no confusing on how we are progressing. We were not able to meet all of our goals during sprint 1 because it was more complicated than we first thought it would be it took a long longer to get the back end connected to the front end than we estimated. This made it difficult to make progress, because the front-end team needed the data from the back end to make the webpages dynamic. We now have the back end connected to the front end and we are making great progress. All things considered, we felt like sprint 1 went pretty well and we only need to make some small changes going forward!

# GitHub Release Link

Create a GitHub release of **a completely working version** of the software and include a link to your release here. Note that some features may not be included if they are scheduled for later sprints, but you must have a **minimum viable product (MVP)** that a user can give feedback on.

Remember to commit to your repository each time a new feature is added/modified. Items should not be marked as “complete” on your backlog until they have been pushed to your repository.

# Product Backlog

Your product backlog contains the list of tasks with their:

* priorities,
* dependencies,
* user story points (an estimate of effort involved from the unitless set {1, 2, 3, 5, 8, 13, 21}), and
* status {not started, in progress, completed}.

It should be updated continually throughout the project. Include the snapshot of the product backlog at the end of the sprint here.

Below are some screenshots of our product backlog in Jira. Each ticket has an assigned number of story points. Each ticket also has its dependencies listed and the backlog is sorted by priorities. The status is tracked by our 3 columns in the sprint board; we have a not started, in progress and completed column.

Text

Description automatically generated

Text

Description automatically generated

Sprint Backlog

Your sprint backlog is a subset of the product backlog. It includes the list of tasks that were scheduled for the sprint. For each task, it provides:

* user story point value (from the product backlog),
* who assigned to,
* actual time spent, and
* status {not started, in progress, completed}.

Sprint backlog:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ticket | Story Point Value | Assignee | Time Spent | Status |
| Learn Django | 8 | Will and Zac |  | In progress |
| Learn React | 8 | Nick and Tarek |  | In progress |
| Update database from python | 3 | Will |  | In progress |
| Connect to database in python | 3 | Zac |  | Completed |
| Request from the database in python | 3 | Will |  | In progress |
| Add button for low item report | 3 | Nick |  | Completed |
| Add button for combo report | 3 | Nick |  | In progress |
| Create page for server GUI | 8 | Tarek |  | Completed |
| Create page for customer GUI | 8 | Cyril |  | In progress |
| Implement order button for server GUI | 3 | Tarek |  | Not started |
| Implement submit button for server GUI | 3 | Tarek |  | Not started |

It should also contain a **summary table** that shows the total number of completed user story points for each team member. The summary table should have a column for each sprint completed to date. These will be a factor in your individual contribution assessment.

# Burn-down Charts

Include two burn-down charts, one for the sprint and one for the product.

* Burn-down charts should be reported in **remaining user story points** (not number of tasks) and show both the user story points remaining and actual hours spent over time. Remember that user story points measure the *relative* estimated effort. They correlate to time spent (but are not equivalent to actual hours/specific units of time.)
* The initial sprint burn-down chart is created from the sprint backlog. It should be updated after each SCRUM meeting based on team member status and feedback. Thus it should contain at least 5 data points (the start of the sprint, 3 SCRUM meetings, and the end of the sprint).
* The product burn-down chart should be updated once during the sprint and again at the end of the sprint. Thus, by the end of the project’s 3 sprints, there should be at least 7 data points on the product burn-down chart (the start of each sprint, the middle of each sprint, and the end of each sprint).

# Next Sprint’s SCRUM Meeting Schedule

Include a list of your planned SCRUM meetings for the next sprint. You should have at least 3 SCRUM meetings in each sprint. (This section does not apply for the final sprint.)

Sprint 2 SCRUM meeting 1: 11/9/22

Sprint 2 SCRUM meeting 2: 11/11/22

Sprint 2 SCRUM meeting 3: 11/14/22

Appendix 1: SCRUM Meeting Agendas and Minutes

SCRUM Meeting 1 for Project 3

Prepared by: Will Thompson

Meeting Date: 11/2/22

## Meeting Attendees

1. Will
2. Nick
3. Zac
4. Tarek
5. Cyril

## Meeting Agenda Items

* Getting everyone’s machine setup with our stack (Django and react.js).
* Assign each member some tickets from the sprint backlog.

## Status Update Since Last Meeting

Accomplishments:

* This is our first meeting so there isn’t anything to update.

Tasks Completed:

|  |  |  |
| --- | --- | --- |
| **Task Description** | **Assigned to** | **Completed? (yes/no)** |
| Setup up Django on everyone’s machines | everyone | yes |
| Assigned tickets to each member | Everyone | yes |

## Before The Next Meeting

Plans:

* Get some simple pages setup on the front end using react
* Start talking to the database using Django

Task Assignments:

|  |  |
| --- | --- |
| **Task Description** | **Assigned to** |
| Create the manager page and add buttons for the reports | Nick |
| Learn Django and connect to the database in Python | Zac |
| Figure out how to request and update the database in Python | Will |
| Learn React and how to create dynamic pages | Tarek |
| Learn about how React and Django connect and keep the group on track | Cyril |

## Minutes from Previous Meeting

This is our first meeting so there is nothing to note for the previous meeting.

SCRUM Meeting 2 for Project 3

Prepared by: Will Thompson

Meeting Date: 11/4/22

## Meeting Attendees

1. Nick Knighton
2. Zac
3. Tarek
4. Will
5. Cyril

## Meeting Agenda Items

* Talk about progress since last meeting
* Assign new tasks as needed so that everyone has something to do

## Status Update Since Last Meeting

Accomplishments:

* Got Django fully setup and connected to the SQL database
* Got React setup so that we can start building the web pages

Tasks Completed:

|  |  |  |
| --- | --- | --- |
| **Task Description** | **Assigned to** | **Completed? (yes/no)** |
| Create the manager page and add buttons for the reports | Nick | No |
| Learn Django and connect to the database in Python | Zac | Yes |
| Figure out how to request and update the database in Python | Will | No |
| Learn React and how to create dynamic pages | Tarek | No |
| Learn about how React and Django connect and keep the group on track | Cyril | No |

## Before The Next Meeting

Plans:

* Continue working on what has been assigned and try to get a minimum viable product
* Get the backend linked to the front end so that we can start doing dynamic actions

Task Assignments:

|  |  |
| --- | --- |
| **Task Description** | **Assigned to** |
| Figure out how to link the front end and back end | Zac and Will |
| Same as last meeting | Nick, Tarek, Cyril |

## Minutes from Previous Meeting

Summarize discussion in paragraph form from the previous meeting (NOT this current meeting).

Last meeting we discussed how to get the project setup so that we could start making progress. We talked about which tech stack we wanted to use. We ended up choosing Django for our back end and React.js for the front end. We then made sure that everyone’s machines where properly setup with everything needed to get started. Overall, the last meeting was a lot of getting started stuff and we didn’t make too much actual progress.

SCRUM Meeting 3 for Project 3

Prepared by: Will Thompson

Meeting Date: 11/7/22

## Meeting Attendees

1. Will
2. Nick
3. Zac
4. Tarek
5. Cyril

## Meeting Agenda Items

* Talk about next steps to get a minimum viable product
* Discuss what we need to get everything submitted for sprint 1

## Status Update Since Last Meeting

Accomplishments:

* Got a simple front end for the manager and server GUIs
* Got the back end linked to the front end

Tasks Completed:

|  |  |  |
| --- | --- | --- |
| **Task Description** | **Assigned to** | **Completed? (yes/no)** |
| Create the manager page and add buttons for the reports | Nick | Yes |
| Figure out how to request and update the database in Python | Will | Yes |
| Learn React and how to create dynamic pages | Tarek | Yes |
| Learn about how React and Django connect and keep the group on track | Cyril | Yes |
| Figure out how to link the front end and back end | Zac and Will | Yes |

## Before The Next Meeting

Plans:

* Get the front end and the back end to communicate
* Add functionality to the front end so that the buttons do something

Task Assignments:

|  |  |
| --- | --- |
| **Task Description** | **Assigned to** |
| Clean up the manager GUI and add a table for the menu items | Nick |
| Work on server GUI | Tarek |
| Work on client GUI | Cyril |
| Get the front end and back talking to each other | Will |
| Learn how to query the database from the back end | Zac |

## Minutes from Previous Meeting

Summarize discussion in paragraph form from the previous meeting (NOT this current meeting).

Last meeting, we had made some small progress from the first meeting. We were able to learn more about how Django and React.js work and how they will end up connecting. The front-end team made some progress on creating the manager and server GUIs. The back-end team was able to get Django connected to React.js so that when we execute the back end, it runs the front end as well. We are now trying to get the back end able to send and receive data from the front end so that we can make the website dynamic.