Resource Manager

Iteration Plan

# 1. Key milestones

|  |  |
| --- | --- |
| **Milestone** | **Date** |
| Iteration Sync Meeting | 12/08 |
| All new implemented activities ready for testing | 23 August 2019 |
| Iteration End | 26 August 2019 |

# 2. High-level objectives

1. Notifications are implemented
2. System users can be appropriately created and removed

# 3. Evaluation criteria

1. users receive notifications
2. Managers can view organisational requirements

# 4. Work Item assignments

The following Work Items will be addressed in this iteration:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Work Item ID** | **Name or key words of description** | **Outcome** | **State** | **Assigned to (name)** | **Estimated Hours** | **Hours worked** | **Estimate of hours remaining** |
| 1.1 | Develop back end components for Request to join project | The back end supports a request to join a project |  | Aaron | 4 |  |  |
| 1.2 | Develop web page for request to join project | The front end can allow a user to request to join a project |  | Jack | 3 |  |  |
| 2.1 | Develop notification back end | The system supports notification of people |  | Matthew | 4 |  |  |
| 2.2 | Implement a notification for a new project assignment | The resources know when they are assigned to a project: Requires 2.1 to be completed |  | Brodie | 2 |  |  |
| 2.3 | Implement a notification for a double booking | Project managers and users know when there is resource contention. Requires 2.1 to be completed |  | Jack | 2 |  |  |
| 2.4 | Develop test cases for notifications | Notifications code is tested against |  | Brodie |  |  |  |
| 3.1 | Implement update system users | Users who are in the system can have their details updated |  | Brodie | 3 |  |  |
| 4.1 | Write full use case description for view organisational requirements | The code can begin development |  | Aaron | 3 |  |  |
| 4.2 | Develop wireframes for view organisational requirements | Requires 4.2 to be complete |  | Aaron | 1 |  |  |
| 4.3 | Develop server code to support view organisational requirements | The Back end can support the view organisational requirements use case. Requires 4.1 and 4.2 to be completed before it can start |  | Matthew | 6 |  |  |
| 4.4 | Implement front end to support view organisational requirements | The front end can talk to the back end to ensure the use case is fully realised |  | Matthew | 3 |  |  |
| 4.5 | Develop test cases for View organisational requirements | The code can be tested. Requires 4.3 and 4.4 to be completed |  | Aaron | 3 |  |  |
| 5.1 | Completed Implementation of update project use case | Carried over from last iteration. Allows the update project use case to be fully realised |  | Brodie | .5 |  |  |
| 5.2 | Complete implementation of Add system users | Adding users to the system is fully functional. | in progress | Jack | 4 |  |  |
| 5.3 | Complete implementation of remove system users | Removing users from the system is fully functional. | in progress | Jack | 4 |  |  |
| 5.4 | Complete implementation for view upcoming projects | Viewing upcoming projects in the system is fully functional. | in progress | Jack | 4 |  |  |

# 5. Issues

|  |  |  |
| --- | --- | --- |
| **Issue** | **Status** | **Notes** |
|  |  |  |
|  |  |  |

# 6. Assessment

|  |  |
| --- | --- |
| Assessment target |  |
| Assessment date | 26/8/2019 |
| Participants | Matthew, Aaron, Brodie, Jack |
| Project status |  |

## Assessment against objectives

## Work Items: Planned compared to actually completed

## Assessment against Evaluation Criteria Test results

## Other concerns and deviations