|  |  |  |
| --- | --- | --- |
| **Phase** | **Item** | **Justification** |
| E-1 | -[Ensure server is suitable and determine most suitable client api](#s74ic4hv3gzb)  -[Implement “Create Account” Use Case](#baltt36zmwmf) | Risk 1  CCRD UC 2 |
| E-2 | -[Implement “Add User To Contacts” Use Case](#8zesitmdd0i1)  -[Implement “View Contacts” Use Case](#egfijtxw9iqa) | CCRD UC 3 |
| E-3 | -[Implement “View Chats” Use Case](#6tz0pjcna64z) [-Implement “Chat” Use Case](#kbfq0km536yi) | CCRD UC 1 |
| E-4 |  |  |
| **UNASSIGNED** | -[Implement “Notifications” Sub-Use Case](#smr4z0edrdl5) | ? Core Architecture ? |
| LCAM (5th June) |  |  |
| C-1 |  |  |
| C-2 |  |  |
| C-3 |  |  |
| C-4 |  |  |
|  |  |  |

**Ensure server is suitable and determine most suitable client api  
Description:**Hello world with openfire server and client api’s (at least Smack, maybe the other two also)  
**Justification:  
[Risk 1]**This is our highest priority risk, and everything else is built on top of this. We cannot afford to spend long not knowing which client api we are using, as this will result in lots of effort being put in the wrong direction.  
**Outcomes:**-Hello world with openfire server the smack api, and hopefully the other two api’s.**Implement “Chat” Use Case**

**Description:**Two users on different client devices can join and participate in a chat.  
Users can start a chat with another user who is in their contacts, by selecting that contact.  
**Justification:  
[CCRD UC 1]**This is our highest priority CCRD UC, as it is the essence of the whole project. We need to show that we can have this basic functionality before moving on to improving it and making the system more usable.  
**Involves:**-Log In  
-Add User To Contacts-View Chats-Send Message  
-Receive Message  
-Message Notification  
**Outcomes:**-A user can start a chat with a contact  
-A message can be sent  
-A sent message is received  
-A user can open an existing chat**Implement “Create Account” Use Case**

**Description:**Users can create accounts (authorization), and then login to their account (authentication).  
**Justification:  
[CCRD UC 2]**This is our second highest priority use case, as it lays the groundwork for fully implementing our highest priority CCRD UC; that is abstracting the creation and entering of chats, via contacts.  
**Outcomes:**-A person can create an account  
-A created account can be logged into from the device that created it and other devices.**Implement “Add User To Contacts” Use Case**

**Description:**Users can add other users via their account. Either by searching for them, or directly by their user handle/id.  
**Justification:  
[CCRD UC 3]**These are our third highest priority CCRD UC, as it is the next step towards fully implementing our highest priority UC.  
**Outcomes:**-A user can search for another user, and request to add them as a contact  
-A user can accept a contact-add request  
-A user can decline a contact-add request**Implement “View Contacts” Use Case**

**Description:**Users can view all their contacts, and search for a specific contact.  
When viewing each contact, they can see their presence.  
**Justification:  
[CCRD UC 3]**These are our third highest priority CCRD UC, as it is the next step towards fully implementing our highest priority UC.  
**Outcomes:**  
-A user can view all of their contacts.  
-A user can view a specific contact.  
  
**Implement “View Chats” Use Case**

**Description:**Users can view all their chats and switch between them.  
**Justification:**This needs to be implemented to some extent during Implement “Chat” Use Case.  
**Outcomes:**-A user can view all of their chats  
-A user can select and view a specific chat from the chats view  
-A user can select and view a specific chat from the contacts view  
  
**Implement “Notifications” Sub-Use Case**

**Description:**Users receive notifications for events such as: contact add request, contact add request accepted, new message.  
**Justification:**Notifications are not part of our CCRD UC’s, however they are an essential part of creating the real time and fast response environment of an instant messenger, and ensuring that important events are not unseen for some time or missed altogether.  
It may also exercise some aspect of the architecture that is not by the other work items.  
**Outcomes:**-Notifications can be displayed on the client