# TODO List Project Plan

## Context:

He team needs a tool to help keep track of all outstanding tasks.

Further information:

* https://github.com/BehaviourLab-Analytics/bl-tech-interview

## Requirements:

Ordered from highest to lowest priority

* Create a UI which allows a user to create, edit and delete a number of TODO tasks.
* The UI only displays the tasks for the currently logged in user.
* Each task has a description inside it which the user can edit.
* The user can assign a status to each item: not started, in progress, paused, complete, abandoned.
* User can edit metadata on items: assigner, project, deadline, priority
* User can order tasks on UI by any metadata field.
* User can filter tasks on UI by any metadata field.
* A user can access their TODO list from any device.

Stretch requirements:

* The user can group tasks together on UI.
* The user can manually reposition and reorder tasks on the UI.
* Items can be moved or copied between user accounts.

## User Stories:

* When I log in, I am taken to the main page showing all open TODO items.
* If I click on new TODO item, I am presented with a form to fill in. Once completed, I am taken to the main page where the new item is shown along with all other open items.
* If I click on filter or sort, the TODO items on the main page rearrange and change according to these selections.

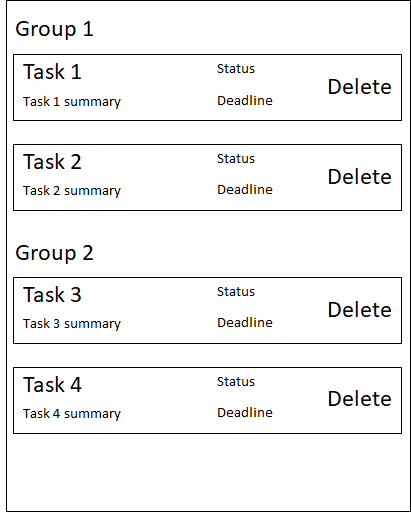
Stretch stories:

* If I drag an item on the main page on top of another, these group together in a clearly displayed separate section.

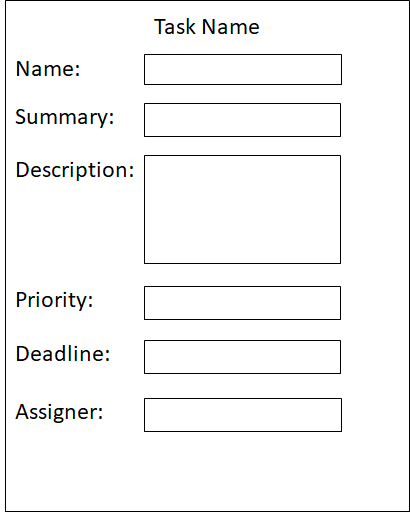
## UI Designs:

These are rough first pass designs and will be subject to discussions and iterations.

### Main page:

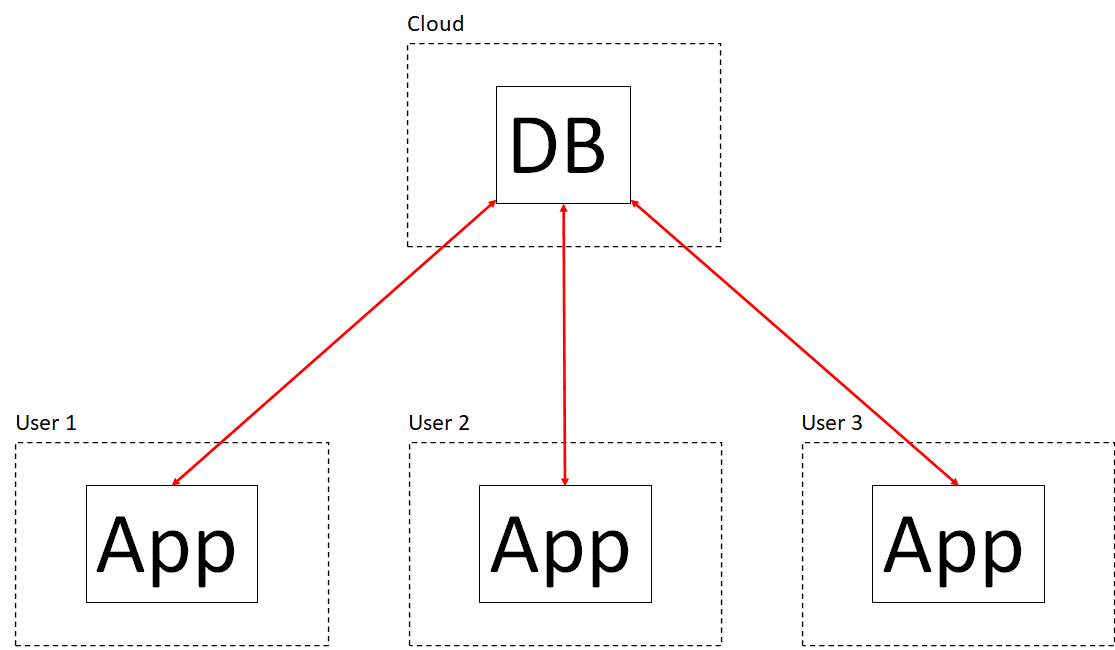


### Create / edit item page:



## System Architecture:

This is the high level system overview, and will be subject to further discussions.



There should be a single cloud based DB which the local applications (one per user) connect to for storing and accessing the data.

## Data Model:

These are the high level data classes and their fields:

### Item:

* ID
* User ID (matches an ID of a User class)
* Name
* Summary
* Description
* Priority (enum)
* Status (enum)
* Deadline (optional)
* Assigner (optional)

### User:

* ID
* Username
* Password (encrypted)
* Last login time

## Use and Scale:

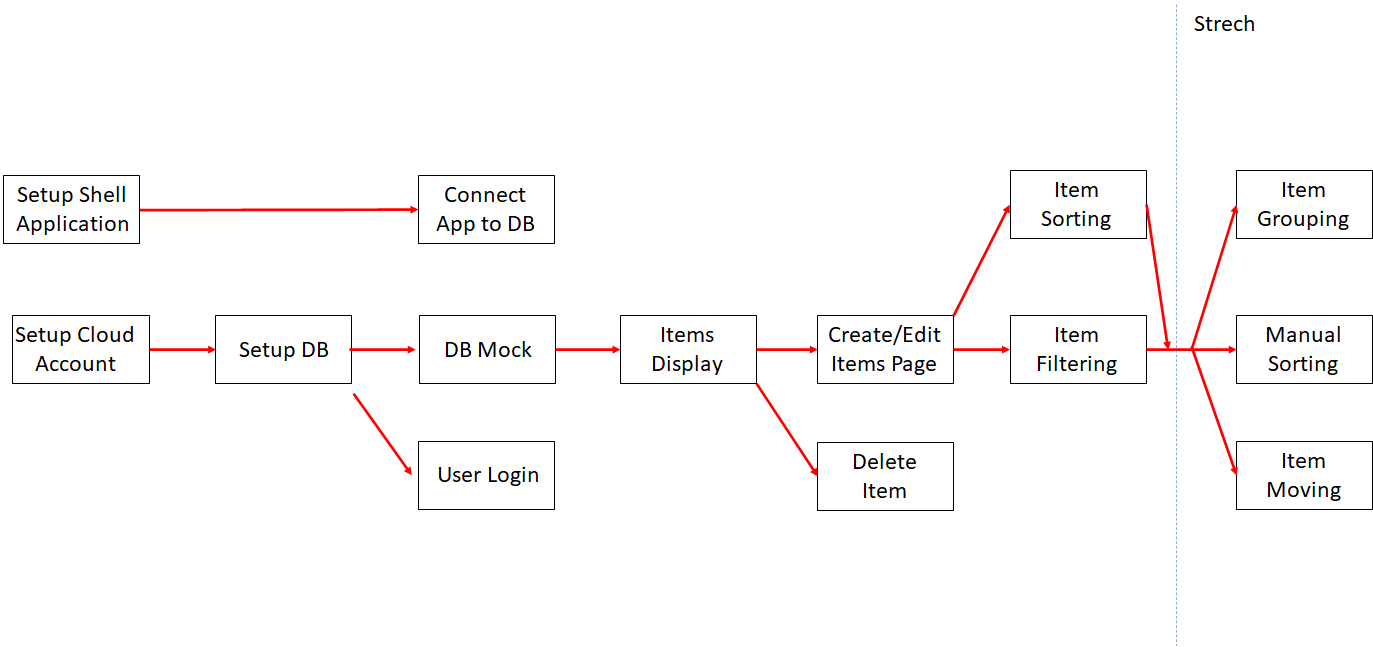
There will be a relatively small number of users (O(tens)) creating a relatively small number of items each (O(hundreds a year)), so the total number of items created per year will likely be O(thousands). The item data model is small so this is not expected to be a large amount of data.

Expected response time for an operation with the DB will be modest (O(tenths of a second)).

Scale is therefore not a great concern in this project.

## Tasks Overview:

An overview of the tasks which will be required. The T-shirt size estimates and any target dates will be determined during discussions with the implementing engineer(s) at the start of development.



* Setup shell application – Create a basic application with the technologies to use. This provides a foundation for all further development.
  + Base UI – Add a blank UI to the shell application.
* Setup cloud account – Create the cloud service which will be used for the DB.
* Setup DB – Setup the DB in the cloud service.
  + Design DB schema - Design the structure of the tables to be used.
  + Create DB tables – Create the tables in the cloud based DB.
* Connect app to cloud DB – Add functionality to the app to read and write to the cloud based DB.
* User login – Allow users to login to the app. This should use as much off the shelf functionality as possible.
* DB mock – Create a DB mick which can be used in local unit testing.
  + Populate mock DB – Add representative test data to the mock DB.
* Items display – Create the UI home page which displays the items.
* Create / edit item page – Add the UI page which allows items to be created and edited.
* Delete item – Add the button which deletes items.
* Item sorting – Add the ability to sort items on the main page.
* Item filtering – Add he ability to filter items on the main page.
* Item grouping – Add the ability to group items on the main page.
* Manual sorting – Add the ability to manually reposition items on the main page.
* Item copying and moving – Add functionality to move or copy items between user accounts.

## Additional Notes:

* The programming languages and technologies to use will be subject to discussions with the implementing engineer(s).