General:

1. ~~Summarize Mark down rules for commenting the code, and provide a document for that. (1 day)~~

~~Link:~~ [~~www.appcoda.com/swift-markdown/~~](http://www.appcoda.com/swift-markdown/)

~~(Issue exists – closed, the links and the description is posted as a comment on the Issue)~~

1. ~~Create the structure of the project, and merge it to DEV.~~

~~(1 day) (Issue exists - done)~~

1. ~~Project Manager: Provide iPhone for testing purposes. (issue exists).~~
2. Create an account to use Siri\_Kit on iPhone (Issue exists, in progress by Simon – update the issue on GitLab with all the information - Katarina).

***First: iPhone workflow:***

1 – Login screen:

* UI team: create the login screen in the main storyboard. (1 day)
  + There is a login screen that Nabid already created (Nabid can explain this in the meeting)
* Issue: Arif, June, Priom  
   Deadline: Thursday
* API: How to authenticate/ de-authenticate the user (Provide a concrete solution). (1 day)

Issue: Nabid  
Deadline: Wednesday at 2pm

* Backend Developer: How to make an HTTP request from XCode. (1 day) (Issue exists)
* Backend Developer: Implement the backend. (2 day)
  + Without the connection to the API
  + Includes creation of the DataBase consisting of Dummy Data

Issue: Katarina, Arian, Mahrukh  
Deadline: Tuesday

* Backend Developer: Implement the backend
  + Connection to the API. (2 day)
* Testers: Are the acceptance criteria met? (1 day)

2- Home screen:

* UI team: Create the home screen in the main storyboard.
* UI team: Create a functional menu icon:
* My Notes: should navigate to (3).
* Logout: just a button without any functionality.

(1 day )

Issue: Arif, June, Priom  
 Deadline: Thursday

* API: How to de-authenticate the user. (refer to 1)

Issue: Nabid

Deadline: Wednesday

* Backend Developer: Implement the logic of the Logout button. (1 day)

Issue: Arian, Katarina   
Deadline: Friday

* Testers: Are the acceptance criteria met? (1 day)

1. My Notes:

* UI Team + Project Manager: Choose how to display the list of notes and what is the look and feel of the edit/delete features (refer to 4 also). (1/2 day)
* UI team: Create the My Notes screen in the main storyboard. (1/2 day)
* API: How to retrieve/update/delete note?
* Backend Developer: Implement the logic of:
* Retrieve the notes and view them. (1 day)
* Delete a note.
* Edit a note (only navigate to 4).

(1/2 day)

* Testers: Are the acceptance criteria met? (1 day)

1. Edit My Notes: (refer to 3)

* UI team: Create the Edit My Notes screen in the main storyboard. (1/2 day)
* API: (refer to 3)
* Backend Developer: Implement the logic of:
* Retrieve the note and view it (make use of logic in 3).
* Edit the note and save it.

(1 day)

* Testers: Are the acceptance criteria met? (1 day)

***Second: CarPlay workflow:***

General:

* Create an app to test the requirements to extend to CarPlay. (1/2 day)

Scenarios:

1. When installing the app, the user must add the shortcuts suggested by the app through the shortcut application.
2. The user has to log in to the app on iPhone.
3. The user must connect the iPhone with CarPlay.
4. To launch the app, the user can:

3-1- Click on the app icon on CarPlay.

3-2- Instruct Siri to launch the app.

3-3- Use any of the voice commands supported by the app (After it is added as a shortcut).

Note: If the user tries to launch the app without being logged in, using one of the former methods, then the response will be:

3-1- The app emits a voice output: “Please log in before using JD Farmer app”.

3-2- The app emits a voice output: “Please log in before using JD Farmer app”.

3-3- Siri will say: “Please log in before using JD Farmer app”.

1. When the app is launched, the app would emit a voice output saying: “Welcome, username, please use Siri to interact with JD Farmer”.
2. User invokes Siri, and gives a voice command (from the shortcut list), and depending on that the app would react.

Possible functionalities:

1. Create a note:

* Voice Command: “JD Farmer take a note”.
* Flow:

1. Siri says: “Ok, Tell me the note?”
2. The user says the note.
3. Siri says: “Ok, this is your note:”, and then repeats the note to the user, then says: “Do you want to save it?”
4. User replies: (Maybe , skip the confirmation part)
   1. “yes”: then the note would be saved by the app, and Siri would notify the user saying: “Your note has been successfully saved”.
   2. “No”: then the note would be discarded by the app, and Siri would notify the user saying: “Ok, the note was canceled”.
   3. Other: Siri replies: “Please reply with Yes or No”.

ToDo: Decide about linking the note to a field, what can be the flow? What if the user gave a name to a non-existing field?

1. Read a note:

* Voice Command: “JD Farmer read my notes [optional parameters]”
* Flow (without optional parameters):

1. Siri says: “Your last note, taken on (date) is” and reads the last note. If there are more notes Siri says: “Do you want to hear more notes?”
2. User replies:
   1. “yes”: Siri reads the note recorded before the note she previously read. If there are more notes Siri says: “Do you want to hear more notes?” then go to step 2.
   2. “No”: Siri says: “Okay”
   3. Other: Siri replies: “Please reply with Yes or No”.

* Flow (with optional parameters):

1. Filter the notes to fit the specified criterion and order them by the time of creation in descending order. Siri says: “Your note is” and reads the last note. If there are more notes Siri says: “Do you want to hear more notes?”
2. User replies:
   1. “yes”: Siri reads the note recorded before the note she previously read. If there are more notes Siri says: “Do you want to hear more notes?” then go to step 2.
   2. “No”: Siri says: “Okay”
   3. Other: Siri replies: “Please reply with Yes or No”.

ToDo: We should decide what are the feasible parameters. This should be discussed with the client(examples: Date).

1. Nearby field updates:

* Voice Command: “JD Farmer give me nearby fields update”.
* Flow:

1. The application would sort the fields in descending order by distance.
2. Siri says: “The (most important ;only for the first update) update of the nearest field is” and reads the update with highest priority. If there are more updates Siri says: “Do you want to hear more updates of this field?”
3. User replies:
   1. “yes”: Siri says the update with lower priority than the previous update said. If there are more updates for the same field Siri says: “Do you want to hear more updates?” then go to step 2.
   2. “No”: Siri says: “Do you want the updates on the other fields?”
      1. User replies:
         1. “yes”: Siri says the update of the next closest field. If there are more updates Siri says: “Do you want to hear more updates?” then go to step 2.a
         2. “No”: Siri says: “Okay”
         3. Other: Siri replies: “Please reply with Yes or No”.
   3. Other: Siri replies: “Please reply with Yes or No”.

ToDo:

* What is the content of the update? This should be given by the API.
* How to prioritize the updates?: This should be discussed with the client.
* UI team: ?. (1 day)
* API: How to create a note? (1 day)
* Backend Developer: Implement the backend. (2 day)
* Testers: Are the acceptance criteria met? (1 day)

Note: use the map to for instance

Color the fields , according to the Last visited time , like red if not visited from 40 days…

Maybe we can show the number of unseen updates