We’d like you to work on the design for a view controller that you get from our **Timesheets** module. The 2 files below represent the same view controller, and the difference is in whether or not you select **PIECE RATE** or **WAGES**.

* [Piece Rate](https://drive.google.com/open?id=1y9nlelo_1PB55XA84GGiTle0hJtjH2jr)
* [Wages](https://drive.google.com/open?id=126T2isJP1Khz6XRqb4vOf1DLhA658NLg)

In there, 02 jobs are represented: Pruning and Thinning. Be mindful of the fact that there can be more than 2 jobs. For pruning you have 2 staff and for thinning only one staff (Again the number of staff here can be variable). For each staff you have:

* Their first name and last name
* The orchard and the block where the staff was clocked in (where the job happened), which were selected on previous screens
* A rate type that you select between Piece rate and Wages
  + If ‘wages’ is selected, you show a label with "job\_name will be paid by wages in this timesheet."
  + If ‘piece rate' is selected, you show a rate field with a button **Apply to all** that, if tapped, will set the same rate to all the other staff under the same job.
* A collection view with some numbers  
  Those numbers represent the rows in a field. When you select a row (so one of those numbers), you get a new label + Tree input view. In this view, you can see something like 2 / 556. 2 represents the current number of trees you pruned or thinned (depending on the job) out of the 556 trees available on the row.   
    
  Sometimes on the row in the collection view you can see a tiny orange dot in the cell. This means that some trees were already pruned on that row from another job. For example for Barco, row 4 has an orange dot and below the text field, there is a label Yi Wan (250). This means that 250 trees out of 556 trees were already pruned through another job on that row. Which means that for the current job you cannot enter more than 556 - 250 = 306 trees.
* Finally, at the top of each job there is an **ADD MAX TREES** button. Clicking this should set the number of trees for each staff, each selected row **under that job** to max acceptable trees.

# Expected Outputs

* A view that looks like the attached designs with the above logic, passing all these [test cases](https://drive.google.com/open?id=1HYaWTo_cZG8gcMGsGoTjHRsp-C6C1P0I) (Previous and next screen is not required. You can open the app on this screen and mock initial data)
* For the list of rows available for selection and the number of already worked trees on it, you can just mock them with an array of object
* Save the data as if you were calling an API
* Add some tests (whatever type you prefer)
* The submission should include (1) a link to your code on GitHub/GitLab and (2) download link for apk.