**What is the scope?**

* Business performance management?

1. If this is the primary goal, capturing financial data becomes very important. This will require talking to other systems already being used in the target organization.
2. It also means that we will be competing with BI tools of established ERP players such as SAP, Oracle etc, as well as standalone players like Tableau in larger companies. In smaller companies this may find more takers
3. Linking to financial performance data also means that prospective clients may not evaluate us until they are close to starting their new financial year, as they would want to start on a clean slate.
4. All the above points made me feel that the saleability of the solution as an off-the shelf, plug and play, batteries-included king of application. I wanted to keep customization to a minimum and people should be able to start using it the moment they buy it. I am open to discussion on this though.

* Goal management?

1. Here the focus is not so much on company’s financial metrics, but on Individuals. For example sales targets, fund raising targets and other objective metrics
2. Goals can still flow from overall business financial targets but the system will not capture financial data from say Tally or SAP
3. This system may end up looking like a normal HRMS, and will attract comparison with numerous other HRMS tools in the market.

* Project & Task Management?

1. Here I am referring to PM in an informal way, not the way it is referred to in large projects which use other sophisticated tools
2. Projects are broken down into tasks and tracked for closure. Various dashboards highlight progress and bottlenecks.
3. May include built in approval forms for travel, purchase, or even in principal approvals

* Collaboration

1. Something like slack
2. I am not keen on getting into this.

Regardless of what the application is trying to manage, the goals / objectives / projects / tasks should cascade from CXO level to the most junior level. In fact as a best practice the application should enforce this by making every goal have a parent goal and child goal (except at the CXO level or the bottom most level where there is not further branching / merging possible). My view is that people can relate to their work more if they have visibility of how their work affects a certain organizational goal.

Again, regardless of the scope chosen, I was planning to include following broad functionalities in the application:

1. Calendar (blocking, planning, visibility of start and end dates of projects/tasks)
2. Messaging
3. Voice Calling (later)
4. Tagging people into a project/task
5. Ability to share attachments from local drive or third party such as Google Drive, Dropbox etc
6. Possible integration with an office suite for preparing excel sheets (eg. For preparing pricing proposals), word documents for quick reports etc. I think integration with google sheets and documents is possible, though I haven’t explored this myself
7. The idea is to stop people from leaving the app frequently to do something else. If they have to replicate data, for example if they are using a CRM system or HRMS where they are already entering some information about what they did during the day/week/month, it will be annoying for them to fill it up again in another app
8. However, more features = more work and considering limited resources and time, I am conflicted on what is the ideal trade off

**UI/UX, Dashboards, Reports**

UI/UX - Application scope needs to be defined first

Dashboards - Some ready made dashboards can be provided. Some ability to customize should be available, but not too much. Same with reports

**Target Segment**

I haven’t thought about this much. Any company with 10 to 1000 employees could be a potential customer. Bigger companies may have separate tools for each type of monitoring. However, not restricting the feature set based on company size. The attempt is to be size neutral.

**Design Philosophy**

People should be able to complete things on the app with minimal clicks. The menu should flow logically with drop down menus and trees, should be uncluttered and should focus on prioritization. It should be the first thing people want to open when they start their work in the morning or the last thing they see when they close for the day. Using some sort of analytics, basic AI and notifications, the app should push people into action and also make them enthusiastic about sharing status updates. There should also be a sense of being part of a community while using the app, but stopping short of looking like a social media circus. I feel this is important when targeting organizations with primarily younger members.

**Recommendation System**

Somewhere down the line I would want to incorporate some sort of AI into the application to make it work like a sort of personal secretary or executive assistant. It should be able to prioritize tasks and customize reports / dashboards automatically based on user activity.

**User Story**

1. Features important to a broad base of users
2. Features important to a small base of important users
3. Velocity, story points, iterations and prioritization. Splitting story points to fit into velocity
4. Writing tests early, if possible even before coding the solution
5. Placeholders for what can be developed later
6. Reducing dependencies between stories to avoid planning and prioritization problems
7. Differentiating between purchaser and user of the software (if such exists)

**User Roles**

1. Users may have different priorities
2. Defining categories of users
3. Brainstorming an initial set of user roles
4. User attributes (e.g. Comfort with using a system, frequency of usage, general objective)
5. Creating a persona (a fleshed out hypothetical user)
6. Considering extreme personas may help in identifying stories we are likely to miss

**Gathering Stories**

1. Get a feel for the big picture before proceeding to write a story in detail
2. See your product in action to see where user is struggling with it
3. Depth first approach. Thrash out one component in detail instead of juggling multiple
4. When there are no users available - check competing products and research what features of their get reviewed most.

**MuSCoW rule**

Must Have, Should Have, Could Have, Won’t have this time