1.

Based on the characteristics of this project, I would suggest using the Scrum Development Methodology. Since:

+ The project has quite many requirements (6 core features).  
+ The requirements may easily to change.  
+ The requirements are not clearly, so they may not determine easily in early stage.  
+ User can appoarch the demo of the project, to provide feedbacks to make the project  
complete

To ensure the project's success, I have created a detailed plan, I will decribe it:

The project has 6 core features, they are:

+ Taks Lists

+ Assignning Tasks

+ Taks Prioritization

+ Deadline Tracking

+ Progress Tracking

+ The module should able to load and display large numbers of tasks without significant delay. User actions should result immediate system responses with minimal lag time. The module should be able to handle a large number of simultaneous users without performance degradatin.

So that I will run 6 sprints for the whole project, each sprint takes 2 weeks, and the scope is to finish 1 feature listed above.

Firstly, our team will create a general plan for the whole project, and design the database using MySQL.

Each sprint is devided into 5 small phases:

+ Phase 1 – Meet and plan (First day of the first week): All members will make a meeting to create a detail tasks list in Jira, review progress, and assign tasks.

+ Phase 2 – Design (about 3-4 days): Our team will design and config the system, config the database as needed, my design team will design the sprint’s UI using Figma.

+ Phase 3 – Code & Test (about 1 week): I and my developer team and my tester team will process programming and testing respectively.

In coding section

- Front-End: Using ReactJS. After finishing several components, our developers will process unit tests and intergration tests.

- Back-End: Using .NET. After finishing some functions and endpoints, our developers will process unit tests and intergration tests.

- Our tester team will process manual test and automation tests.

- Our QA (Quality Assurance) team will ensure the quality of the project.

+ Phase 4 - Release: The demo version of the project would be released to the customer for review and feedback.

+ Phase 5 – Feedback: The customer would provide feedback on the demo version, and the team would make any necessary adjustments before proceeding to the next sprint.

By utilizing the Scrum methodology and breaking down the project into manageable sprints with clear phases, the team can work efficiently and effectively to create a high-quality product that meets the needs of the customer

2.  
List four functional requirements:

+ Task Lists: allow team members to proritize tasks and organize their work in a way that makes sense to them.

+ Assigng Tasks: allow the leader to give tasks to themselves or orthers. This helps ensure that tasks are distributed evenly across the team and that everyone knows what they need to do.

+ Deadline Tracking: allows the leader to set deadlines for tasks and to track the progress of charges againts those deadlines.

+ Progress Tracking : track the progress of tasks and communicate with other team members about task status. This helps ensure that everyone is on the same page and that there are no surprises regarding task completion.

List two non-functional requirements:

+ Capacity: Maximum concurrent users. For example, the system should be able to handle 10,000 concurrent users.

+ Performance: Response time. For example, the system should respond to user input within 2 seconds.

3. User stories

* As a user, I want to check my schedule for working, so that, I can caculate my salary
* As a user, I want to search my tasks by date and by weeks, so that, I can check tasks and my deadline to avoid missing tasks.

4. Story map

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Assigning Tasks | Task Prioritization | Deadline Tracking | | Progress Tracking | |
| Giving tasks | Prioritize tasks | Set deadlines for tasks | Track the progress of charges | Track the progess of tasks | Communicate with others |
| Send a taks | Rate the importance and urgency | Select time and date | View progress status | View tasks status | Send the messages |
|  | Send the prioritizes |  |  |  | Make a call |
|  |  |  |  |  |  |

5. List three assumptions regarding the deadline-tracking feature.

+ Set wrong deadline for tasks: High impact if wrong, High Probability of it being wrong

Explain: There are many members may not work at same locations and time zones, which make more difficult to developer to complete a function, also easy to make confuse when the leader set the deadline. At a result, the productivity of working will be decreasing beasce they did not finish the deadline at expected.

+ Set deadline for the wrong person: High impact if wrong, low Probability of it being wrong

Explain: Some time the leader will confuse because of large number of members, and set the deadline for wrong person. The consequences realy serious but it can avoid because each member will have things to indentities themsevles.

+ Wrong in showing the progress of charges against those deadlines: High impact if wrong, Low Probability of it being wrong.

Explain: If there is some wrong in caclulating, it will showing wrong progress when leader try to track the progress of charges, which is dangerous because, the leader cannot contact with member on time when they get in trouble making theirs tasks.

6. The kind of testing that I suggest to my team are:

+ Unit Testing: Unit Testing, which involves testing the smallest isolated piece of code in the system. For Front-End testing, I suggest using Jest(Javascript), while for Back-End testing, I suggest using Swizzar(.NET).

+ Intergation Testing: We will test the software modules and their logical integration as a group. Again, I suggest using Jest(Javascript) for Front-End testing and Swizzar(.NET) for Back-End testing.

+ Automation Testing: Automation Testing will increase the effectiveness and speed of testing. Our team use this method because it saves time and money, eliminates the need for manual testing of all workflows and negative scenarios, and increases test coverage. Additionally, Automation Testing allows for unattended testing, which increases the speed of test execution, and helps to reduce errors that may occur during manual testing.