Hello everyone. In this part, I will introduce the software product of our project.

Since our product targets to normal customers, offering an intuitive and beautiful user interface is our top priority. Featuring existing libraries and our own enhancements, we provide a consistent look, rich and interactive charts and animations, all of which lead to a fluent and comfortable user experience. It’s a mobile world, so we use responsive design to ensure that you can use our system with any devices you have, including mobile phones, tablets and large screen PCs. We have a homepage as an introduction of our product. We have detailed dashboard with figures, tables and charts all presented in a harmonious way. When you use your phone, the page adjusts to a mobile friendly view. You can also resize the screen as you wish and see how our page adapts to it.

Under the hood, a maintainable and extensible software powers the system.

We separate frontend and backend. This enables parallel and specialized development, debugging, testing and optimization since they are technically isolated and don’t involve another when changed. It also helps in network performance and the integration of other clients since the data needed to be transferred are minimized and well defined.

As you can see, API is the bridge between two ends, and therefore the most important part in our system. It needs to be precisely documented. With the help of Swagger, we write APIs as code, and Swagger automatically generates docs based on them so that the docs are always up to date.

In backend, we create a layered architecture with specialized layers which also helps development, debugging and testing. We choose Flask as the web framework for it is light weight, easy to use and has a rich ecosystem which brings convenience into development process. Also, since most machine learning tools provides interfaces for Python, it’s easy to integrate them into our project.

In frontend, we create Single Page Application with React, which is the most welcomed frontend framework in the world and has a large and evolving community. We believe that a good code quality is the key to a robust, maintainable and extensible software no matter in frontend or backend. So, we use various tools to ensure the quality of our product. TypeScript with TSLint makes us write elegant code in an elegant way; Code splitting splits huge mono package into small chunks to achieve load on demand which reduces unnecessary network traffic; Complex state management is made orderly and easy with MobX; Dependency injection results in a loosely coupled, extensible and maintainable codebase. We also create API layer so that during development we can dynamically choose whether to use mock data or not by just changing a variable, which is extremely convenient and time-saving in practice.

Software development is not a one man’s fight, so we stress teamwork and form a modern workflow to ensure the success of our project. We use git as version control and based on that, we have a hierarchical branch system and a branch-based workflow. First, we write our code in one’s own branch. It will be merged into frontend or backend when it’s completed and needs review and test. When a feature is completed on both ends, they are merged into software for an integration test, and when it’s ready to be released, we merge software into master. This process ensures that our work is organized properly and orderly.

That’s all for software part. We will play a demo video after the final chapter is finished. Thank you!