# Estimation

## Database

1. Database schema preparation 4-8h
2. Tool for automatic updating database 8h

## Environment

1. Setting up environments: dev, test, prod 16-24h
2. Setting up git repository, CI software, version manages (ex .octopus) 24h
3. Preparing build scripts 16-24h
4. Code versioning 16h

## API

1. Endpoints for movie list components 16h
2. Checkout Component 24-32h
3. Rental Component 24-32h
4. Customer Management Component 16-24h
5. Other: API versioning, IoC etc. 16h

## Web

1. Endpoints for movie list components 16h
2. Checkout Component 24-32h
3. Rental Component 24-32h
4. Customer Management Component 16-24h

## Testing

1. Unit test, functional test 80-104h

# Solution in Repo

1. Solution in repo contains only part of API project. It’s prototype:

* No validation
* No tests
* Mocked data

1. Picture below show proposed architecture. So basically we would create two project. One of the would be web application (single page app) create in react with redux or in angular. And second would be API project created in ASP.NET Web API 2, which would communicate with database and serves the data. I decided to have separate because API is easier to scale. We don’t need teens of WEB project, but for sure having couple API project would make all system work faster. Additionally to that I decided to group functionalities in API project into components so that if they grow larger they can be easily extracted into another service and scale separately.

Why those technologies?

I picked Redux because it makes us think of our application as an initial state being modified by a sequential list of actions, which I found is a really nice way to approach complex web apps and opens up a lot of opportunities. React allows us to build html components, which are easy to maintain and it works cool with redux. As for the API technologies I love .net technologies, I have a fondness for them. The solution what Microsoft provides are really reliable.

C:\Users\Arkadiusz.Stachowiak\Downloads\Untitled Diagram (1).png