1. Login page
2. Map/Grid View
3. Multiplayer: Chats
4. Background Updates
5. Cooldowns
6. Database Design

Everything takes time to build, fight, share resources, gain resources -> Clock

Entities:

1. Player
   1. Login
2. Settlement:
   1. Main Home
   2. Colonys: limieten op resources
3. Building:
   1. Resources: Woodcutter, Stonecutter, Farmer, Metalforger
   2. Policitcs: Castle , Storage
   3. Attack: Trainingscamp
4. Guild/Clan/Group
   1. Chat met elkaar
   2. Resources uitwisselen

Wood + Stone = Building

Metal + Time = Soldier

Soldier eats food over time. If not enough food; Soldier dies

Fight System: Different types of soldiers with its own health & damage. Hit button fight; select troops. Simulate fight in background.

Technisch:  
Database

Flask

Frontend

Auto Build & Deploy

Taakverdeling:

- Game Design: (Jonas)

- Cooldowns : ratio

- Visualisation: hoe ziet een entity eruit?

- Lore

- Scope Description uitschrijven in 1 do

- Frontend

- Login Page

- Grid/Kaart

- Chat/Social Menu: DM & GroupChat (Guilds)

- Guild Overview

- Build Menu

- Material Menu

- Fight Page

- Settlement

* API werkend maken

- API kunnen schrijven

- Connecten aan Databse & Frontend

- Backend

- Data storen van users login

- API voor Login

- Store van standaard info

- Resource gestored per user

- Database Design

- Database ER Diagram

* Database Run Scripts

- Server Setup (Kars)

- Auto Build (Kars)

Time: Keep Status of Player -> By login; update everything, while logged in: update per 5 seconden

15:15; Ik val Jonas aan met 5 Speermannen. Dat duurt 30min voordat mijn soldaten bij Jonas zijn. Om 15:45: simuleer fith: 5 speermannen vs Jonas Soldiers. En dan om 15:15: message Kars & Jonas met report over fight: who won, which soldiers died -> Fix time priorities