**Dokumendihaldussüsteem**
Ülesande püstituseks on luua veebirakendus, kus oleks võimalik näiteks väikefirmadel ametlikke dokumendipõhju hoida ja täita: otsused, aktid, lepingud, protokollid jne
Dokumendipõhju saavad kasutajad omavahel jagada ning kiirelt täita vormi abil.

Kuna tegu on projektiga väikesele tiimile, siis iga tiimi liige peaks üldjoontes olema kursis kõigi süsteemi osadega (full stack).

Süsteem koosneb järgmistest osadest:
\* Front end - Single Page Application (SPA) kasutades Reactjs, VueJS vm.

\* Vajadusel back end teenus(ed), mis suhtlevad front endiga üle REST API

\* Andmebaas dokumentide hoidmiseks (Mongo, firebase, dynamodb)

\* Autentimine - (amazon cognito, google oauth)

Näitlikud projekti sammud agiilse lähenemisega:
1. Panna paika esmane plaan, otsustada süsteemi arhitektuur ja tehnoloogiad, mida kasutada (võivad hiljem muutuda)

2. Leppida kokku projekti skoop ja featuurid, mille tegemine tundub projekti raames reaalne.

3. Teha Kanban tahvel, näiteks Trellos ja rivistada sinna olulisuse järjekorras ülesanded. Tahvli lahtrites on näiteks: backlog, to do, doing, done. Ülesanded lisatakse backlogi alati.

4. Ülesannete jaotus hakkabki toimuma kanbani abil. Tiimi liikmed võtavad järjest ülesanded, liigutades nad todo alla, mis näiteks nädala (sprindi) jooksul plaanis ära teha on.

5. Ülesanne mis parasjagu käsil, liigutatakse 'doing' lahtrisse.

6. Kui ülesanne sprindi lõpuks valmis, siis liigub see done alla. Kui ei, siis läheb edasi järgmisesse sprinti. Sellisel juhul võib-olla ülesanne liiga suur ja saab teha väiksemateks alamülesanneteks, millega tiimi kaaslased ka aidata saavad.

7. Kanban tahvel annab ülevaate projekti kulgemisest ja näitab ära, kus on peamised pudelikaelad.

8. Esimene prioriteet on saada projekti skelett püsti. Näiteks kuna meil on tegu veebirakendusega võiks see olla näiteks react boilerplate projekt, mida saab iga tiimi liige kohalikult oma arvutis jooksutada.

9. Kindlasti kasutada versioonihaldustarkvara, soovitatavalt git-i. <https://gitlab.pld.ttu.ee/>

Iga tiimi liige peaks ideaalis arendama iga kanbani pileti eraldi branchis, kui see valmis siis võiks toimuda pull request, tiimi liikmed koodi üle vaadata ja seejärel mergeda masterisse.

10. Keskenduda alguses MVP (minimal viable product) loomisele. Alati peaks mingi töötav seis projektist olemas olema. Kui leiduvad valmis komponendid, mis nõudmisi rahuldavad, pole tarvis jalgratast leiutada.

Antud projekti puhul põhilised ülesanded, mis peaksid omakorda alamülesanneteks jaotuma:

  1) Dokumendi redigeerimine

  2) Dokumendi laadimine ja salvestamine kohalikult.

  3) Dokumentide hoiustamine andmebaasis

  4) Dokumentide nimistu loomine

  5) Autentimine, dokumentide sidumine kasutajatega.

  6) Dokumentide täitmise võimalus vormi abil (kuupäev, lepingu osaliste nimed jms)

  7) Dokumendi allalaadimise võimalus PDFina

11. Deployment (valikuline)