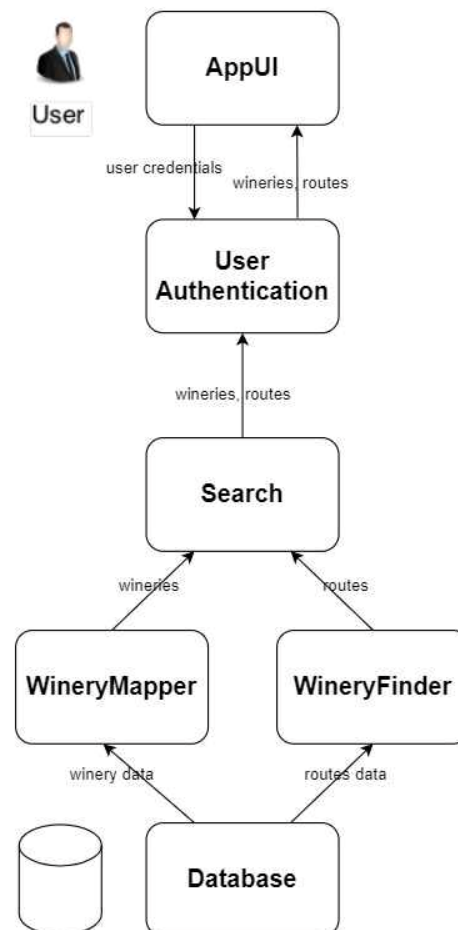


1. CONCEPTUAL MODEL



1.1. Conceptual model

APP UI:

- Show the graphical interface
- Show map
- Show object

User Authentication:

- Handles user login and ensures only authorized individuals enter.
- Checks credentials and grants access to app resources.
- Manages user access and keeps unauthorized users at bay.

Search:

- Show every object (winery) that matches the keyword attribute

Winery Finder:

- Search by winery name (exact or partial match), location (city, town, region), occupation (type of wine produced, e.g., red, white, sparkling).

Winery mapper:

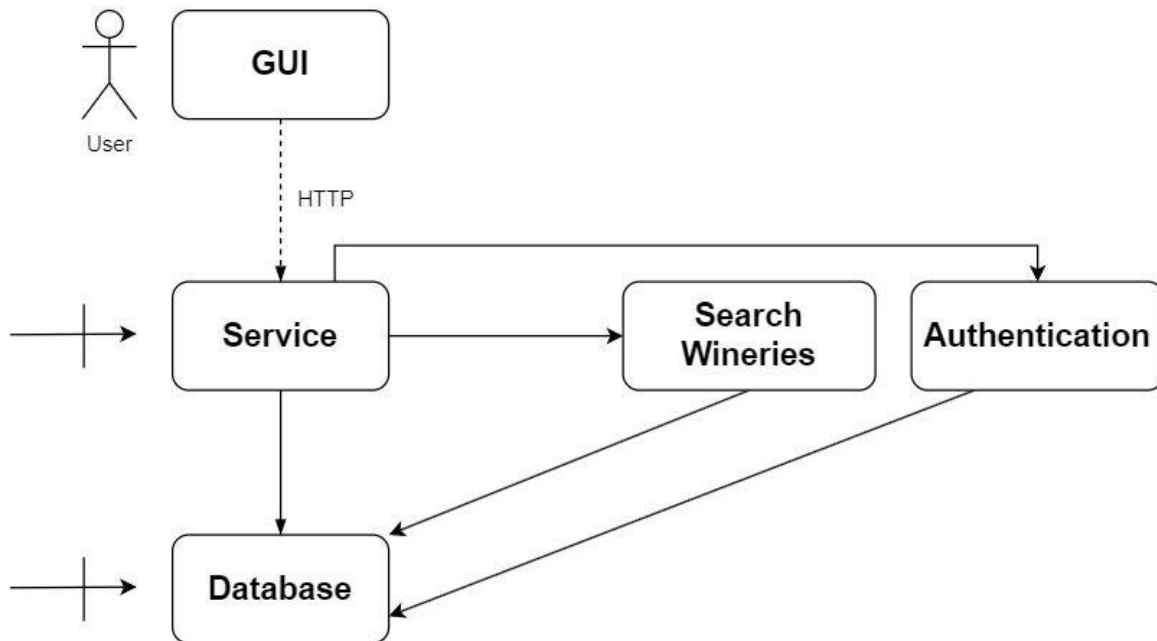
- Display an interactive map of R. Macedonia

- Display detailed information about each winery when clicked on.
- Calculate and display directions from the user's current location to the selected winery.

Database:

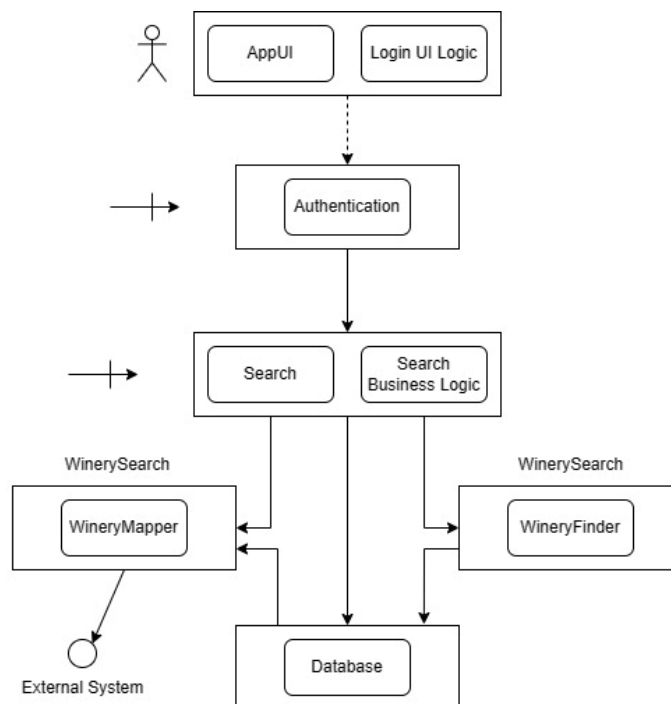
- Stores winery data, user profiles, wishlists, and routes.

2. EXECUTION MODEL

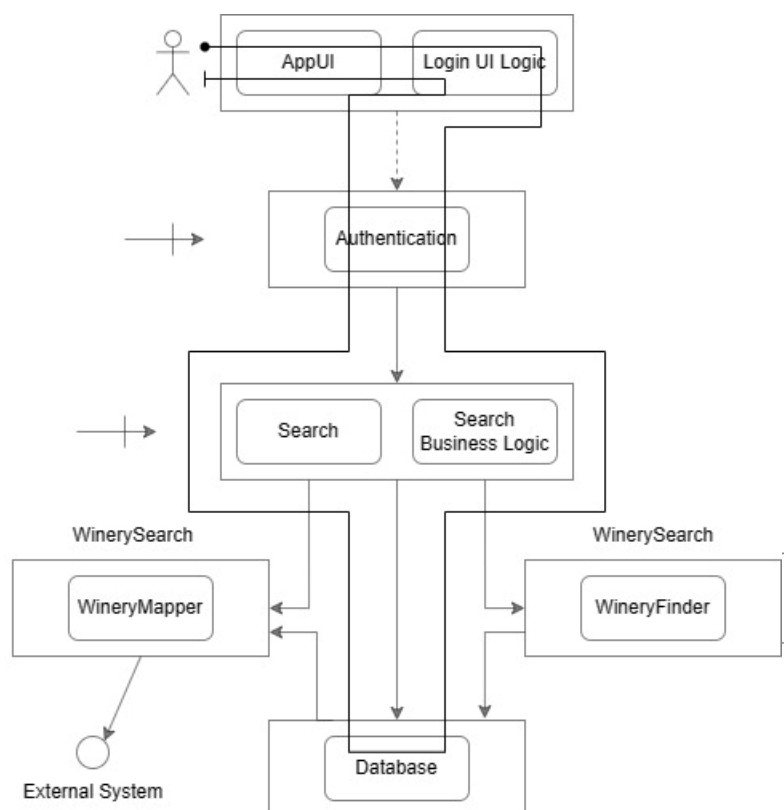


2.1. Execution model

- На корисникот му е прикажан GUI (graphic user interface) преку кој тој комуницира. Преку соодветен мрежен протокол испраќа барање до Service за автентикација. Потоа Authentication Service ги потврдува ингеренциите на корисникот и дава пристап до услугата за пребарување. Услугата за пребарување потоа ја пребарува базата на податоци за винарии и ги враќа резултатите на корисникот.

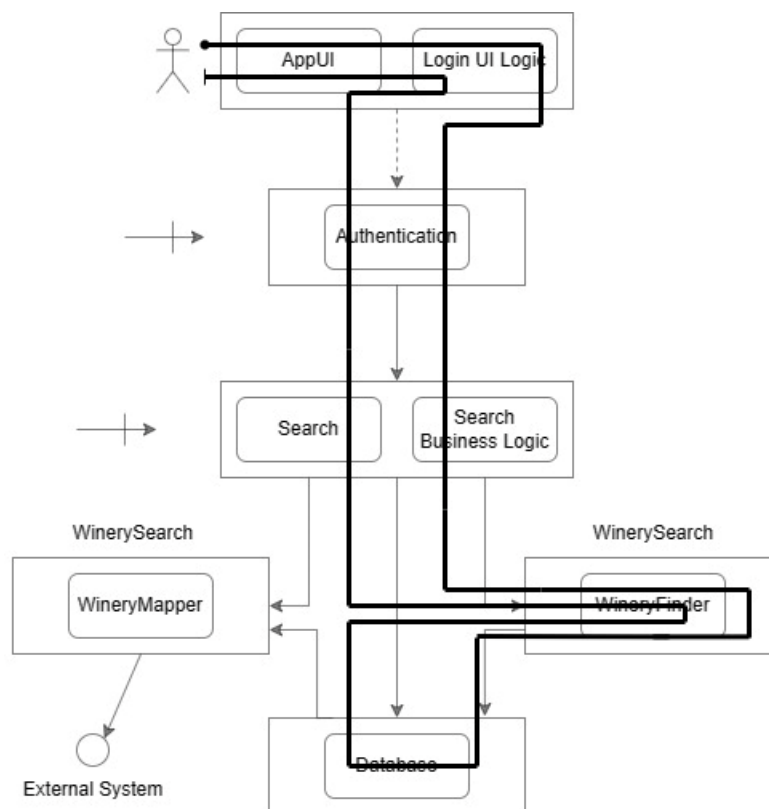


2.2. Conceptual and execution model



2.3. Conceptual and execution search authentication model

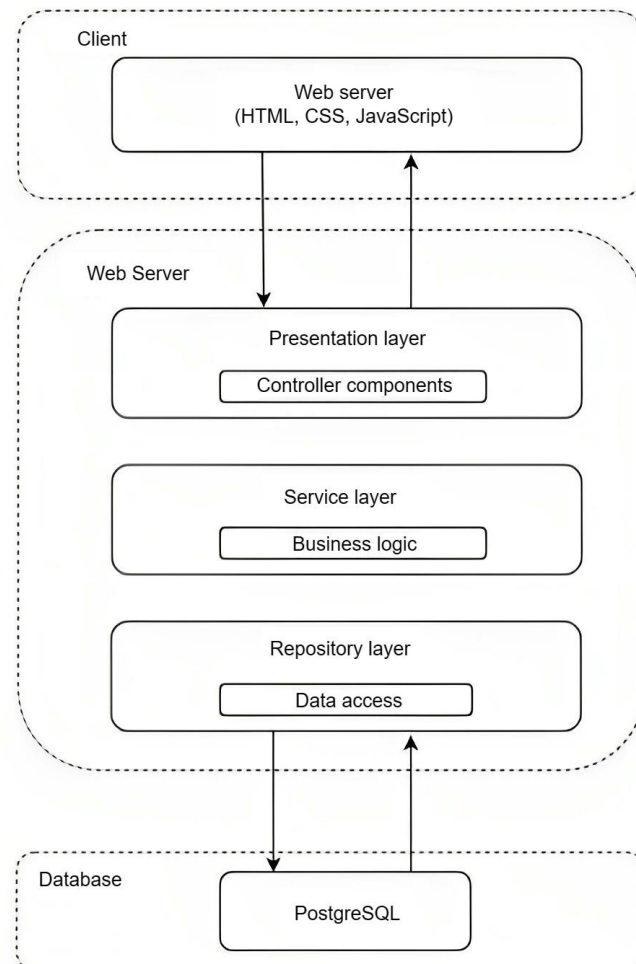
- При логирање на корисник , корисничкиот интерфејс испраќа податоци до Authentication кој го насочува барањето со податоците внесени од корисникот до базата на податоци во која се чуваат информации за оние корисници кои се регистрирани. Проверува дали има совпаѓање и доколку има му дозволува на корисникот пристап за користење на услугите од апликацијата. Доколку не го пренасочува повторно кон интерфејсот за логирање.



2.2. Conceptual and execution search behavior model

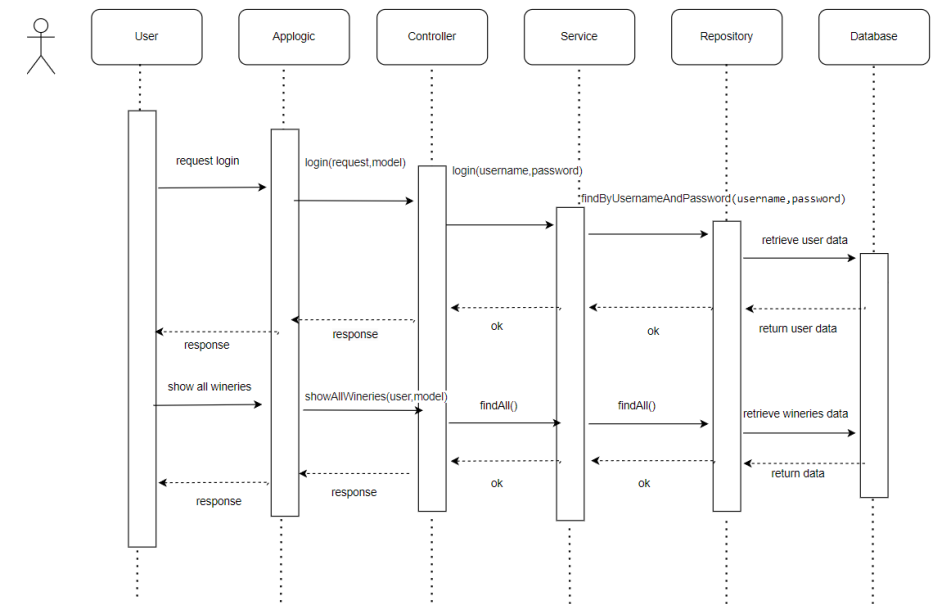
- При пребарување на винарија, корисникот најпрво мора да биде автентизиран. Барањето испратено од корисничкиот интерфејс најпрво се проверува дали е испратено од автентизиран корисник. Истото барање се проследува до Search Business Logic која податоците испратени во барањето ги препраќа на WineryFinder задолжено за пребарување на винарии според клучни атрибути. WineryFinder ги селектира винариите од базата на податоци според клучните зборови и повторно ги враќа да се прикажат на корисникот како листа на винарии кои се совпаѓаат со пребарувањето.

3. IMPLEMENTATION MODEL



3.1. Implementation model

- При барање на клиентот, му се прикажува на корисникот страница на пребарувачот која содржи HTML, CSS, JavaScript код. Клиентот комуницира со веб серверот преку презентацискиот слој, од каде се примаат и праќаат барањата кон клиентот. Сервисниот слој ја врши бизнис логиката на барањата, тој е поврзан со презентацискиот слој од една страна и со податочниот слојот од друга страна. Податочниот слој комуницира со базата на податоци и од неа ги повлекува потребните податоци кои се потребни да ги прати на сервисниот слој.



3.2. Sequence diagram

- Корисникот на системот може да направи барање со цел да ги добие сите винарии. Најпрво имаме логирање на корисникот, кој доколку е автентичиран ќе може да направи барање. Овде е претставено барање, во кое корисникот сака да добие приказ на сите винарии. Од базата на податоци како одговор ќе се вратат сите винарии.

*Во гореспоменатите дијаграми и архитектури, секаде каде што се користи поимот база на податоци или Database се мисли на база на податоци која ќе биде имплементирана во PostgreSQL, но поради тоа што во моментот на имплементација тоа не е користено, се користи генерализацијата Database.