

Архитектурни стилови и дизајн

1. Концептуална (концепциска) архитектура

Дијаграмот за концептуална архитектура е поедноставен приказ и може да не ги опфати сите интеракции или детални компоненти. Овој дијаграм го нагласува протокот на барања и одговори помеѓу frontend, backend и базата на податоци. Идни дополнителни функционалности, може да се вклучат во дијаграмот по потреба. Како прелиминарен чекор, ги испитавме функционалните и нефункционалните барања, нагласувајќи ги клучните концепти и термини како што е илустрирано во табелата подолу.

Data	Function	Stakeholder	System	Abstract
Локации на винарии	Приказ на навигација	Корисник	FastAPI Backend	Интерактивна мапа
Информации за винарии (име, адреса, итн.)	Приказ на мапа (интеграција на OpenStreetMap)		Vue.js Frontend	Повеќејазична поддршка
“System Backup” податоци	Пребарување винарии		External services	Лента за навигација
Навигациски линкови	Филтрирање винарии			
	Приказ на пребарувањето и филтрирањето			

1.1. Компоненти на архитектурата

1.1.1. Кориснички интерфејс:

1. Веб прелистувач: Примарниот интерфејс за корисниците да комуницираат со системот.
2. Лента за навигација: Содржи копчиња за приказ на мапа и контакт
3. Приказ на мапа: Прикажува карта со интегриран OpenStreetMap и локации на винарии.

1.1.2. Логички слој на апликацијата:

1. Јазична поддршка: Спроведува повеќејазична поддршка (македонски и англиски).
2. Секција за пребарување и преглед: Се справува со кориснички податоци за пребарување на винарии врз основа на име, адреса, град и активност.
3. Интеграција на мапа: Се поврзува со OpenStreetMap API за прикажување на локации на винарии.
4. Интерактивна мапа: Обезбедува динамична мапа со значки за секоја винарија, овозможувајќи им на корисниците да кликнат за детални информации.

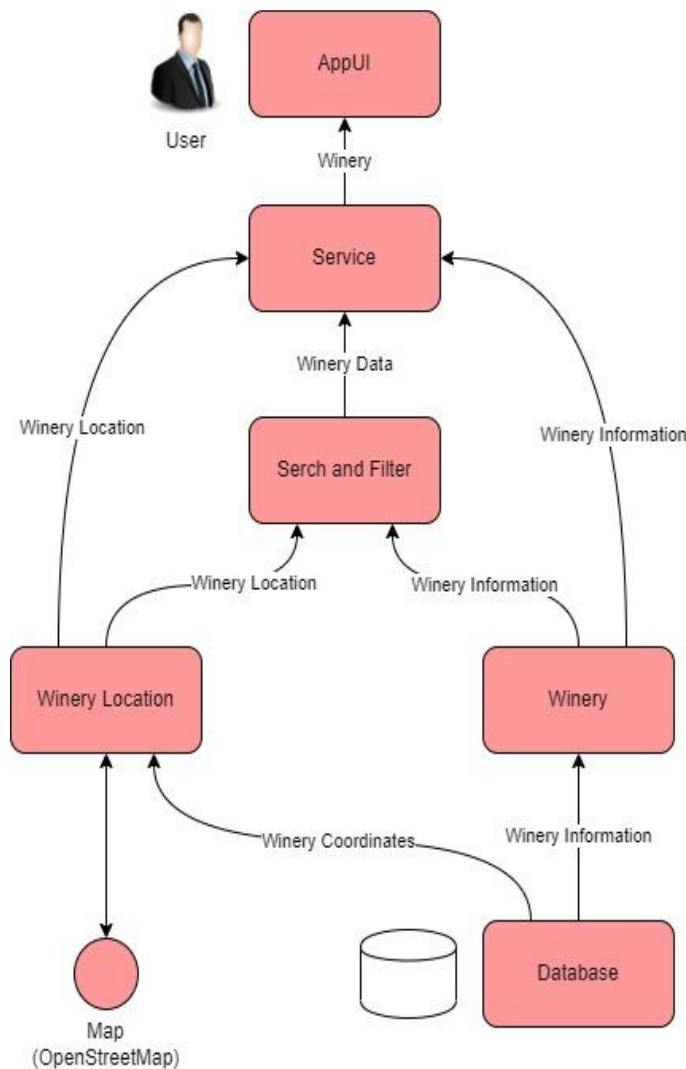
1.1.3. Слој на податоци:

1. База на податоци за винарии (CSV File): Чува информации за винарии, вклучувајќи име, адреса, град, активност, социјални мрежи, контакт информации, работно време, како и координатите на локацијата.
2. Врзување со база на податоци: Преку File I/O API се чита и ажурира базата.

1.1.4. Микросервиси:

1. Услуга за винарија (Winery MS): Управува со операциите CRUD за информации за винариите (во овие операции припаѓа и Search and Filter).
2. Услуга за мапи (Winery Locator MS): Се интегрира со API на OpenStreetMap за превземање и прикажување на мапа и локации.

1.2. Слика од концептуална архитектура



2. Извршна архитектура

2.1. Компоненти на архитектурата

2.1.1. Веб архитектура:

- Layered Architecture: Ги раздвојува проблемите на UI, апликациска логика, податоци и слоеви на микросервис.
- Pipe and Filter: За обработка на влезни податоци од страна на корисникот и прикажување резултати.

2.1.2. Микросервиси:

- Услуга за винарија (Winery MS): Се справува со функциите поврзани со информациите за винариите.
- Услуга за мапи (Winery Locator MS): Се справува со функциите поврзани со мапата и локациите.

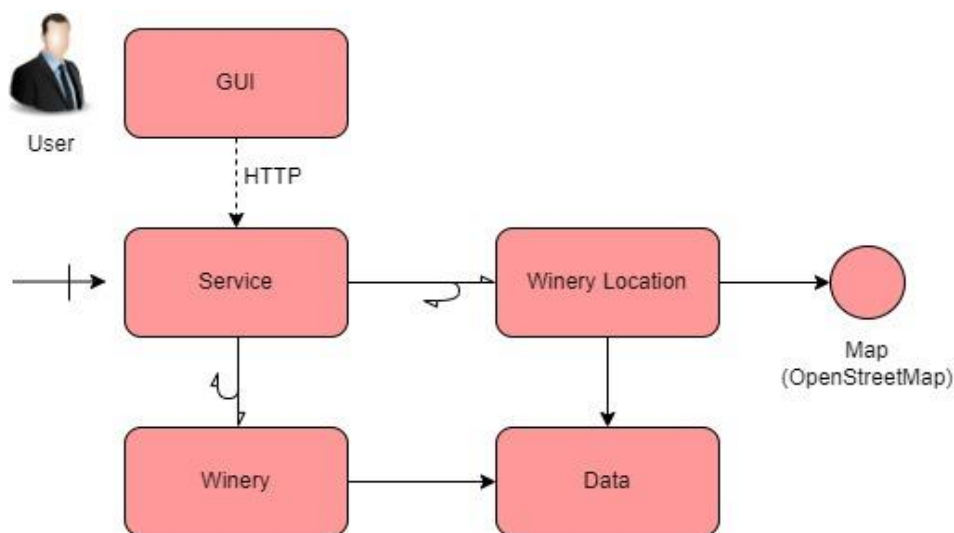
2.1.3. Контејнеризација:

- Docker Containers: Ја пакува секоја микроуслуга во контејнери за лесно распоредување и приспособливост.

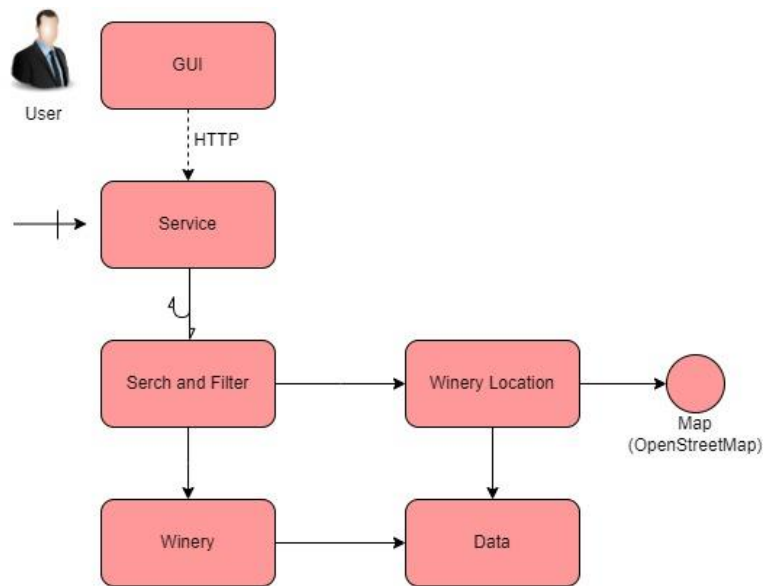
2.1.4. Протоколи за комуникација:

- RESTful API: За комуникација помеѓу frontend (Vue.js), FastAPI и микросервисите. Ова ја стандардизира комуникацијата.

2.2. Слика од извршна архитектура



Втор поглед, Слика 2.



Прв поглед, Слика 1.

2.2.3. Тек на извршување:

- Service ги презема и ги прикажува сите достапни локации на винариите од Data (CSV датотека).
- Корисниците комуницираат со GUI, кој може да вклучува функционалности обезбедени, како што е Search and Filter.
- Service комуницира со микросервисите Winery и Winery Locator за ажурирање на прикажаните информации за винаријата врз основа на критериумите за пребарување и филтрирање.

3. Имплементациска архитектура

3.1. Компоненти на архитектурата

3.1.1. Програмски јазик и рамки:

- Frontend: HTML, CSS, JavaScript (Vue.js)
- Backend: Python (FastAPI)

3.1.2. Микросервиси:

- Креирање контејнери со помош на Docker

3.1.3. Интегрирање на мапа:

- Приказ на мапа со помош на OpenStreetMap API

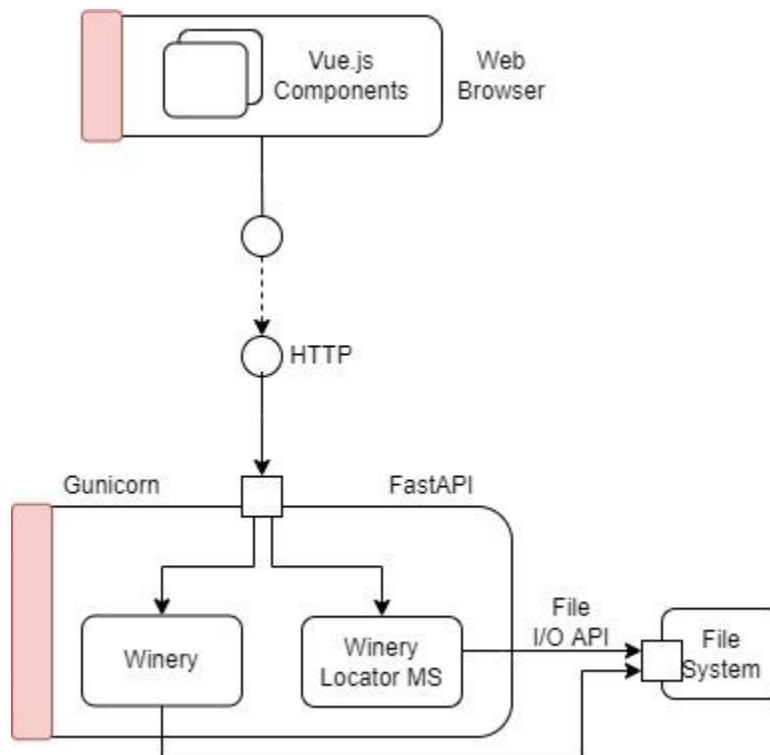
3.1.4. База на податоци:

- Сите информации за винариите се чуваат во CSV File

3.1.5. Безбедност:

- SSL/TLS: Обезбедува сигурна комуникација помеѓу веб апликацијата и корисниците.

3.2. Слика од имплементациската архитектура



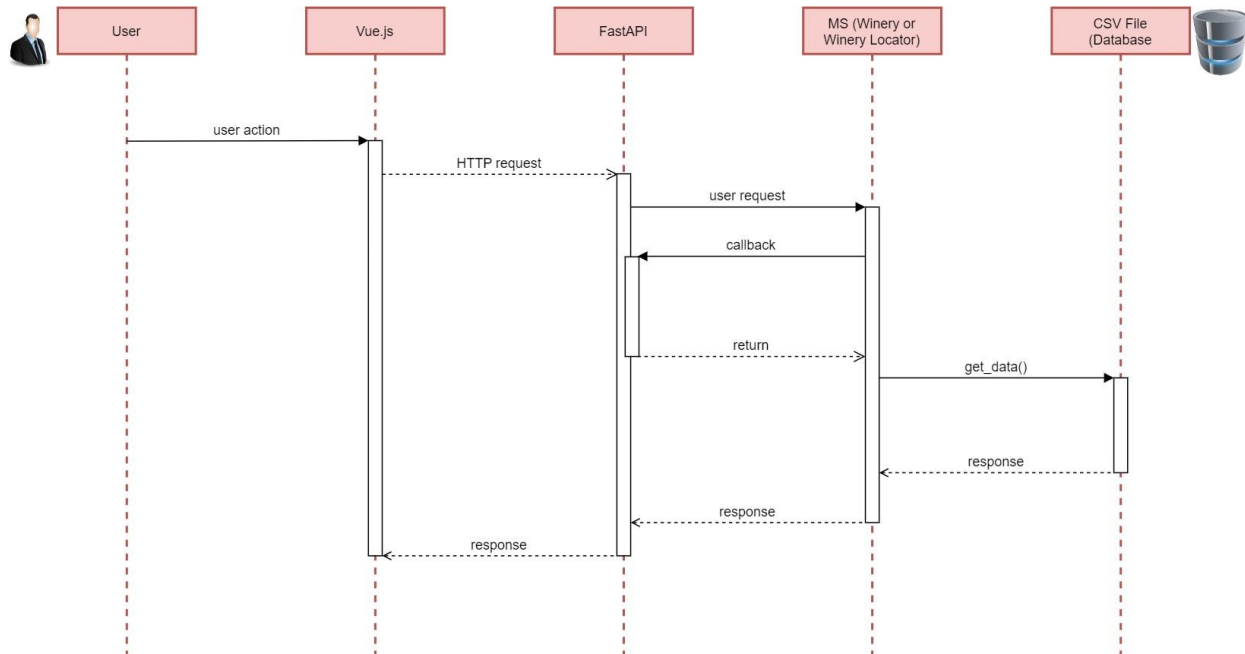
3.2.1. Тек на извршување:

- Корисниците комуницираат со frontend делот на Vue.js, иницирајќи барања за податоци.
- FastAPI ги прима овие барања, ги обработува и комуницира со базата на податоци за да преземе или ажурира информации за винариите.
- FastAPI ги испраќа обработените податоци назад во Vue.js за прикажување.

3.2.2. Додатни информации:

- Vue.js и FastAPI комуницираат преку HTTP барања (RESTful API).
- FastAPI се справува со деловната логика, вклучувајќи операции за пребарување и филтрирање.
- Базата се чите преку File I/O API поради тоа што е CSV File, а не SQL база.
- Базата на податоци се чита и се ажурира од backend делот врз основа на дејствата на корисникот.

3.3. Секвенценцијален дијаграм (Behaviour model)



3.3.1. Објаснување на секвенцијалниот дијаграм:

- Корисничка интеракција со компонентите на Vue.js: Корисникот комуницира со компонентите на Vue.js, како што се копчињата или полињата за внесување, иницирајќи корисничко дејство.
- API повик од Vue.js до FastAPI: Компонентите на Vue.js прават API повик до FastAPI, испраќајќи барање за HTTP. Овој повик го вклучува дејството на корисникот и релевантните податоци.
 - Мрежен протокол HTTP/HTTPS: API повикот помеѓу Vue.js и FastAPI го следи мрежниот протокол HTTP/HTTPS за да обезбеди сигурна комуникација.
- FastAPI повик до микроуслуги: FastAPI, по добивањето на повикот за API од страна на Vue.js, го насочува барањето до соодветните микроуслуги (Winery и Winery Locator).
 - API повик од FastAPI до Winery Микросервис: FastAPI прави API повик до Winery, означувајќи ја конкретната операција што треба да се изврши. Овој повик ги вклучува дејството на корисникот и релевантните податоци.
 - Сигнал за повратен повик од FastAPI до Winery: Има сигнал за повратен повик од FastAPI до Winery, што укажува дека микросервисот треба да го обработи барањето.
 - Директен повик од Winery до CSV базата на податоци: Микросервисот директно ја повикува базата на податоци користејќи го File I/O API за преземање или ажурирање информации за винариите.
 - API повик од FastAPI до WineryLocator Микросервис: Истовремено, FastAPI прави API повик до Winery Locator, означувајќи ја конкретната операција што

треба да се изврши. Овој повик исто така ги вклучува дејството на корисникот и релевантните податоци.

- Сигнал за повратен повик од FastAPI до Winery Locator: Има сигнал за повратен повик од FastAPI до Winery Locator, што укажува дека микросервисот треба да го обработи барањето поврзано со локацијата.
 - Директен повик од Winery Locator до CSV базата на податоци: Микросервисот директно ја повикува базата на податоци користејќи го File I/O API за за операции поврзани со локација.
4. API повик од FastAPI до компонентите на Vue.js: Микросервисите ги завршуваат нивните соодветни операции, а FastAPI ги консолидира податоците. FastAPI потоа испраќа API повик до компонентите на Vue.js, одговарајќи со обработените податоци. Повикот исто како и предходно го следи мрежниот протокол HTTP/HTTPS за безбедна комуникација.
 5. Ажурирање на компонентите на Vue.js: Компонентите на Vue.js го добиваат одговорот на соодветниот API повик и го ажурираат корисничкиот интерфејс врз основа на обработените податоци.