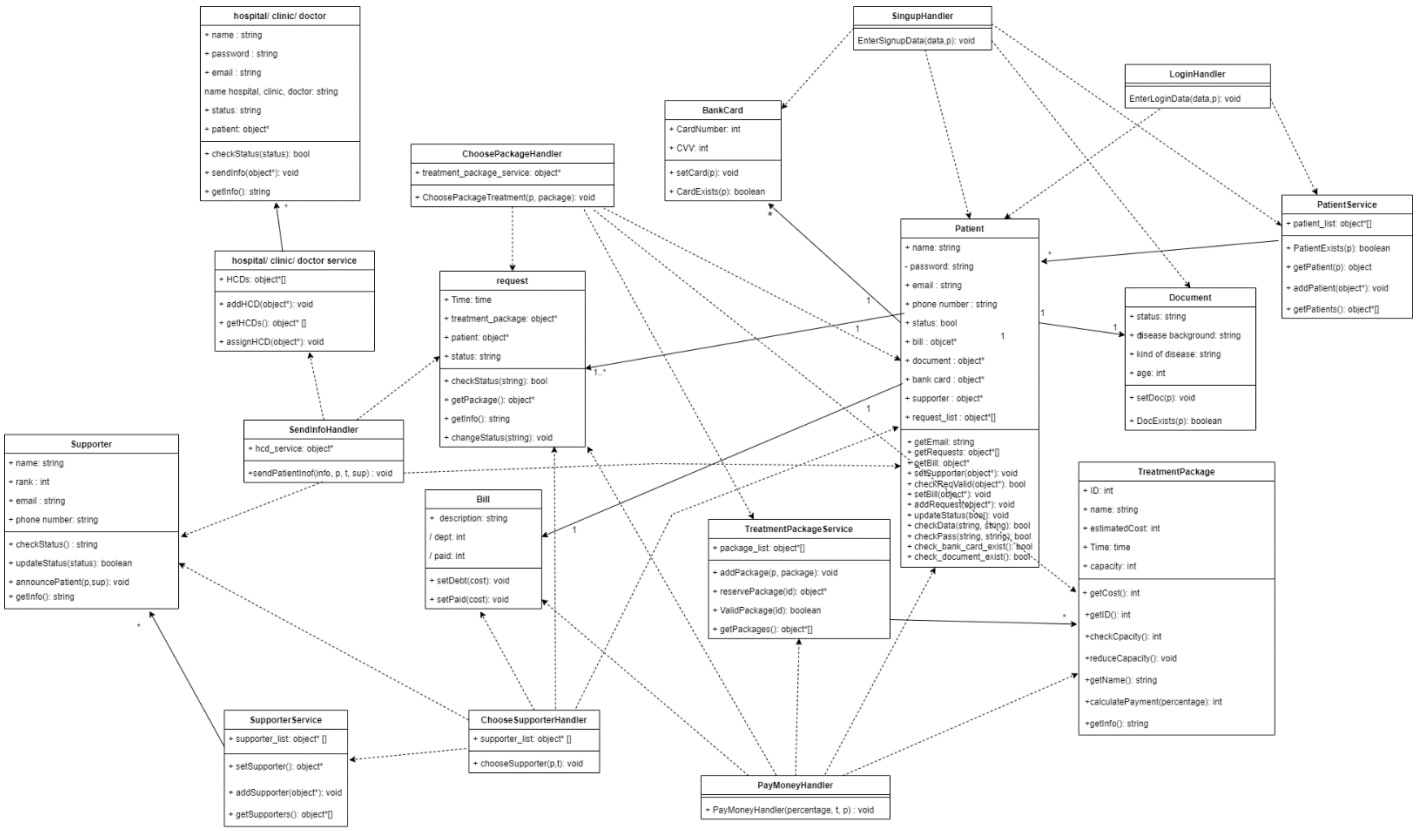
تحلیل و طراحی سیستم ها
توریست درمانی: فاز پنجم

سنا ساری نوایی: 810199435
مریم جعفرآبادی آشتیانی: 810199549
اولدوز نیساری: 810199505

در این پروژه قصد داریم کلاس های طراحی شده برای object های برنامه را پیاده سازی کنیم. برای اینکار از زبان cpp استفاده کرده ایم. در هنگام پیاده سازی متوجه شدیم لازم است برخی قسمت های طراحی نمودار کلاس در فاز قبلی را تغییر دهیم. نسخه بروز شده آن در شکل زیر آمده است.



حال به بررسی کلاس های پیاده سازی شده در کد میپردازیم و دلیل وجود هریک را بررسی میکنیم:

- BankCard

در این کلاس نمونه ای از کارت بانکی ساخته میشود و پوینتری از آن در نمونه ای از قبل موجود از کلاس بیمار ذخیره میشود.

- Bill

در این کلاس نمونه ای از صورتحساب ساخته میشود و پوینتری از آن در نمونه ای از قبل موجود از کلاس بیمار ذخیره میشود. ساخت صورتحساب توسط کلاس ChoosePackageHandler صدا زده میشود. این

تحلیل و طراحی سیستم ها
توریست درمانی: فاز پنجم

سنا ساری نوایی: 810199435
مریم جعفرآبادی آشتیانی: 810199549
اولدوز نیساری: 810199505

کلاس مقدار پرداختی و مقدار بدهی بیمار مشخص است. وقتی بیمار مقدار 25 درصد پیش پرداخت را پرداخت میکند، مقدار بدهی و پرداختی را اپدیت میکنیم.

- ChoosePackageHandler

این کلاس برای هندل کردن انتخاب بسته درمانی ساخته شده است. درواقع به نوعی ارتباط میان بیمار و بسته درمانی را مدیریت میکند. وقتی بیمار بسته درمانی را انتخاب میکند، ابتدا بررسی میشود که بسته درمانی انتخاب شده موجود باشد. سپس ظرفیت آن بسته بررسی میشود. در نهایت اگر تمامی موارد گفته شده مشکلی نداشتند، آن پکیج برای بیمار موردنظر رزرو میشود. مقدار Bill برای بیمار تنظیم میشود و مقدار هزینه بسته درمانی به بدهی های بیمار افزوده میشود.

- ChooseSupporterHandler

این کلاس برای انتخاب کردن پشتیبان سلامت مناسب برای بیمار موردنظر میباشد. بعد از پرداخت پول توسط بیمار، روند انتخاب پشتیبان سلامت صورت میگیرد. در این کلاس ابتدا چک میکنیم که درخواست وجود داشته باشد و همچنین آن درخواست تایید شده باشد. در صورتیکه موارد گفته شده رعایت شده باشند، انتخاب پشتیبان سلامت انجام میشود. پس از انتخاب پشتیبان سلامت، استاتوس آن در حالت busy قرار میگیرد و استاتوس درخواست نیز در حالت sup assigned قرار میگیرد که به معنی این است به بیمار، یک پشتیبان سلامت تخصیص داده شده است.

- Document

این کلاس مربوط به پرونده بیمار است و در آن اطلاعاتی نظیر بیماری زمینه ای، سن، نوع بیماری و ... ذخیره میشود. پوینتری از آن در نمونه ای از قبل موجود از کلاس بیمار ذخیره میشود. ساخت پرونده در کلاس SingupHandler صورت میگیرد.

- HDC_Service

این کلاس اطلاعات مربوط به تمامی بیمارستان/کلینیک/دکتر ها را ذخیره میکند. عملیات اضافه کردن بیمارستان/کلینیک/دکتر نیز در این کلاس انجام میشود. اگر بیماری در یک بیمارستان/کلینیک/دکتر رزرو کند، استاتوس آن از حالت available خارج میشود و به حالت busy میرود.

- HospitalClinicDoctor

این کلاس مربوط به بیمارستان/کلینیک/دکتر میباشد. یعنی اطلاعات هر یک از بیمارستان/کلینیک/دکتر نظیر اسم کارشناس بیمارستان/کلینیک/دکتر ، ایمیل کارشناس و نام خود بیمارستان/کلینیک/دکتر قرار دارد. اطلاعات بیمار توسط پشتیبان سلامت برای کارشناس بیمارستان/کلینیک/دکتر فرستاده و در این کلاس ذخیره میشود.

تحلیل و طراحی سیستم ها
توریست درمانی: فاز پنجم

سنا ساری نوایی: 810199435
مریم جعفرآبادی آشتیانی: 810199549
اولدوز نیساری: 810199505

- LoginHandler

این کلاس برای هندل کردن عملیات لاگین تعریف شده است. وقتی بیمار گزینه لاگین را انتخاب میکند، ابتدا چک میشود که از قبل ثبت نام کرده باشد. سپس استاتوس آن را اپدیت میکنیم و در حالت true قرار میدهیم. قبل از لاگین نیز باید چک کنیم که بیمار کارت بانکی و پرونده نیز داشته باشد. در صورتیکه تمامی موارد گفته شده رعایت شده باشد و بیمار اطلاعات را صحیح وارد کرده باشد، بیمار لاگین میشود و میتواند از امکانات سایت استفاده کند.

- Patient

در این کلاس اطلاعات مربوط به هر بیمار نظیر نام، پسورد، ایمیل، شماره همراه و ... قرار میگیرد. اکثر توابعی که در این کلاس قرار دارند، برای چک کردن وضعیت یا داده ای مربوط به بیمار است. مثل checkPass() ، check_bank_card_exist() و غیره. توابع دیگری نیز مانند getBill() ، getEmail() و موارد مشابه برای این به کار برده شده اند که اطلاعات مربوط به بیمار را بتوانیم دریافت کنیم. پس از آنکه بیمار ثبت نام کند، یک کلاس patient برای آن بیمار ساخته میشود و اطلاعات بیمار در آن کلاس قرار میگیرد.

- PatientRequest

این کلاس مربوط به درخواست بیمار میباشد. وقتی بیمار یک بسته درمانی را انتخاب کند، این بسته صدا زده میشود. استاتوس آن ابتدا در حالت not confirmed قرار میگیرد و وقتی کار مربوط به آن بسته درمانی انجام شود، استاتوس آن تغییر میکند.

- PatientService

در این کلاس اطلاعات تمامی بیماران نگهداری میشود. هدف از ایجاد کلاس این بوده است که راهی برای دسترسی به همه بیماران فراهم شود. وقتی بیمار لاگین میکند، باید چک کنیم این بیمار از قبل ثبت نام کرده است یا خیر. برای این کار از ایمیل بیمار استفاده میکنیم. به ازای تمامی بیماران ثبت نام شده، ایمیل فردی که درخواست لاگین داده است را با بیماران دیگر چک میکنیم. در صورتیکه برابر باشد، در آنصورت عمل لاگین موفقیت آمیز خواهد بود.

- PayMoneyHandler

در این کلاس هندل کردن پرداخت پول صورت میگیرد. وقتی بیمار بسته درمانی موردنظرش را انتخاب میکند، باید مقدار 25 درصد پول را به صورت پیش پرداخت بپردازد تا درخواستش ثبت نهایی شود. برای اینکار ابتدا چک میکنیم که درخواست بیمار ثبت شده باشد و استاتوس آن در حالت not confirmed قرار گرفته باشد. پس از آن عملیات پرداخت انجام میشود. مقدار بدهی و پرداختی بیمار را در کلاس

سنا ساری نوایی: 810199435
مریم جعفرآبادی آشتیانی: 810199549
اولدوز نیساری: 810199505

صورتحساب ذخیره میکنیم. استاتوس درخواست به حالت confirmed تغییر می یابد زیرا درخواست بیمار پس از پیش پرداخت نهایی میشود.

- SendInfoHandler

این کلاس برای هندل کردن بیمار و بیمارستان/کلینیک/دکتر میباشد. در این کلاس، مشخص میشود که بیمار در کدام بیمارستان/کلینیک/دکتر درمانش را انجام میدهد و همچنین پشتیبان سلامت به بیمار پیامی ارسال میکند و اطلاعات را به او میدهد.

- SignupHandler

در این کلاس ثبت نام بیمار صورت میگیرد. اطلاعاتی که از بیمار دریافت میکنیم را در کلاس بیماری که تعریف کرده ایم، ذخیره میکنیم. قبل از ثبت نام، چک میکنیم بیماری با اطلاعات داده شده داریم یا خیر. اگر وجود داشته باشد، ارور میدهیم و عملیات ثبت نام صورت نمیگیرد. اگر اطلاعات درست باشد، یک کلاس پرونده بیمار و یک کلاس کارت بانکی مربوط به بیمار نیز درست میشود و بیمار را نیز به کلاس PatientService اضافه میکنیم؛ در این کلاس همانطور که گفتیم، اطلاعات تمامی بیماران نگهداری میشود.

- Supporter

این کلاس مربوط به پشتیبان سلامت است. اطلاعات مربوط به پشتیبان سلامت مانند نام، رنک، ایمیل، شماره همراه و استاتوس در این کلاس ذخیره میشود. توابعی که در این کلاس استفاده شده اند، برای اپدیت کردن یا گرفتن اطلاعات از این کلاس استفاده میشوند. مانند ()checkStatus که استاتوس را برمیگرداند یا تابع AnnouncePatient(patient) که بیماران مربوط به پشتیبان سلامت را در وکتوری ذخیره میکند.

- SupporterService

در این کلاس پشتیبان سلامت مناسب را برای بیمار انتخاب میکنیم. این کلاس لیست تمامی پشتیبان های سلامت را در خود ذخیره دارد. به این صورت عمل میکنیم که از میان تمامی پشتیبان های سلامت، استاتوس آنها را بررسی میکنیم. اگر در حالت busy قرار نداشت، در اینصورت آن پشتیبان سلامت تایم خالی دارد و بیمارمان را به او اختصاص میدهیم. انتخاب پشتیبان سلامت بعد از مرحله واریز پیش پرداخت میباشد.

- TreatmentPackage

این کلاس مربوط به بسته های درمانی میباشد و اطلاعاتی نظیر آیدی، نام، هزینه تقریبی و ... در آن قرار میگیرد. توابعی که در این کلاس تعریف شده اند، برای تغییر دادن برخی اطلاعات بسته درمانی هستند.

تحلیل و طراحی سیستم ها
توریست درمانی: فاز پنجم

سنا ساری نوایی: 810199435
مریم جعفرآبادی آشتیانی: 810199549
اولدوز نیساری: 810199505

به عنوان مثال تابع reduce_capacity() وقتی به کار میرود که بیمار بسته درمانی ای را انتخاب کند. در اینصورت باید از ظرفیت آن بسته یک واحد کم کنیم. همچنین تابع calculatePayment(int percentage) برای محاسبه مقدار پیش پرداختی است که بیمار باید پرداخت کند.

- TreatementPackageService

این کلاس تمامی بسته های درمانی را در خودش دارد و میتوان بسته درمانی به آن اضافه کرد یا اطلاعات مربوط به یک بسته درمانی مشخص را دریافت کرد. همچنین در این کلاس رزرو کردن بسته درمانی نیز صورت میگیرد. ابتدا آیدی بسته درمانی انتخاب شده توسط بیمار را دریافت میکنیم و با جست و جو در بین تمامی بسته های درمانی، بسته موردنظر را پیدا میکنیم. اگر بسته درمانی ظرفیت داشت، از ظرفیت یک عدد کم میکنیم و آن بسته را به بیمار تخصیص میدهیم. در صورت نداشتن ظرفیت، پیام ارور به کاربر نشان داده میشود.

توضیح برخی قسمت های مهم کد

```
try
{
    MyServer server(5000);
    server.get("/", new ShowPage("html/main.html"));
    server.get("/main_css", new ShowPage("html/css/main.css"));
    server.get("/icon", new ShowImage("html/images/icon.png"));
    server.get("/main_background", new ShowImage("html/images/mainpage.jpg"));
    server.get("/background", new ShowImage("html/images/main2.avif"));
    server.get("/show_signup", new ShowPage("html/signup.html"));
    server.get("/signup", signup_handler);
    server.get("/show_login", new ShowPage("html/login.html"));
    server.get("/login", login_handler);
    server.get("/show_package", new ShowPage("html/package.html"));
    server.get("/midwifery", new ShowImage("html/images/midwifery.jpg"));
    server.get("/orthopedics", new ShowImage("html/images/orthopedics.jpeg"));
    server.get("/General", new ShowImage("html/images/General surgery.jpg"));
    server.get("/ENT", new ShowImage("html/images/ENT.jpg"));
    server.get("/diabet", new ShowImage("html/images/diabet.jpg"));
    server.get("/kidney", new ShowImage("html/images/kidney.jpg"));
    server.get("/package", choose_package_handler);
    server.get("/show_paymoney", new ShowPage("html/payMoney.html"));
    server.get("/payMoneyBackground", new ShowImage("html/images/moneyBackground.jpg"));
    server.get("/paymoney", pay_money_handler);
    server.get("/supporter", choose_supporter_handler);
    server.get("/sendInfo", send_info_handler);
    server.get("/Thank_show", new ShowPage("html/thanks.html"));
    server.get("/thankyou", new ShowImage("html/images/thankyou.jpg"));
    server.run();
}
catch (const Server::Exception &e)
{
    std::cerr << e.getMessage() << std::endl;
}
```

تحلیل و طراحی سیستم ها
توریست درمانی: فاز پنجم

سنا ساری نوایی: 810199435
مریم جعفرآبادی آشتیانی: 810199549
اولدوز نیساری: 810199505

در این قسمت پیاده سازی اولیه سرور صورت گرفته است. پورت 5000 را برای آن انتخاب کردیم. برای کنترل آن هندلر های متفاوتی در نظر گرفته ایم کا کارکرد تمامی آنها در بخش قبل توضیح داده شده است. همچنین برای بارگذاری صفحات سایت و تمامی فایل های عکس و غیره نیز در این بخش صورت میگیرد. اگر یک exception پیدا شود، پیام خطا مرتبط با آن به کاربر نمایش داده میشود.

```
PatientService *patient_service = new PatientService();
SignupHandler *signup_handler = new SignupHandler(patient_service);
LoginHandler *login_handler = new LoginHandler(patient_service);
TreatmentPackageService *treatment_package_service = new TreatmentPackageService();
ChoosePackageHandler *choose_package_handler = new ChoosePackageHandler(treatment_package_service);
PayMoneyHandler *pay_money_handler = new PayMoneyHandler();
SupporterService *supporter_service = new SupporterService();
ChooseSupporterHandler *choose_supporter_handler = new ChooseSupporterHandler(supporter_service);
HCD_Service *hcd_service = new HCD_Service();
SendInfoHandler *send_info_handler = new SendInfoHandler(hcd_service, supporter_service, treatment_package_service, patient_service);

DAO *dao_layer = new DAO(patient_service, treatment_package_service, supporter_service, hcd_service);
dao_layer->read_data_base();

OneStepToTreatment *ourWebsite = new OneStepToTreatment(patient_service, signup_handler, login_handler, treatment_package_service,
                                                        choose_package_handler, pay_money_handler, supporter_service,
                                                        choose_supporter_handler, hcd_service, send_info_handler);
ourWebsite->run();

dao_layer->update_data_base();
```

تابع main() نیز بخش مهم بعدی محسوب میشود. در این تابع ابتدا از تمامی کلاس های موجود یک instance گرفتیم و سپس آنها را OneStepToTreatment پاس داده ایم. قبل از آن، DAO layer قرار دارد که مسئول تعامل با سیستم ذخیره سازی مانند پایگاه داده یا سیستم فایل است. از این لایه برای عملیات CRUD استفاده میشود. قبل از اینکه کاری صورت بگیرد، با استفاده از DAO اطلاعات را از دیتابیس میخوانیم و پس از پایان کار نیز اطلاعات را در فایل اپدیت میکنیم.