

دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

**طراحی سیستم رزرو زمان ویزیت بیماران ساختمان پزشکان**

استاد راهنما :

جناب آقای دکتر پوروطن

اعضای گروه :

آریا بنائی زاده . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9431029

حامد حیدریان . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9431058

محسن صادقی گرمارودی . . . . . . . .. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9431063

سید نوید کرمی نژاد . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9431070

محسن رضوانی . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9431803

فهرست مطالب

فهرست مطالب

[تعریف پروژه 3](#_Toc487250902)

[1. مقدمه 3](#_Toc487250903)

[1.1. هدف 3](#_Toc487250904)

[2. تعریف اولیه 3](#_Toc487250905)

[3. تعریف کلی پروژه 3](#_Toc487250906)

[4. نیازمندی‌ها 3](#_Toc487250907)

[4.1. بیمار 3](#_Toc487250908)

[4.2. پزشک 5](#_Toc487250909)

[4.3. مدیر سیستم 6](#_Toc487250910)

[4.4. فاعل زمان 7](#_Toc487250911)

[Structured Analysis 8](#_Toc487250912)

[1. Use Case 9](#_Toc487250913)

[1.1. Use Case Diagram 9](#_Toc487250914)

[1.2. Use Case Glossary 10](#_Toc487250915)

[1. Priority Table 24](#_Toc487250916)

[2. Entity Relationship Diagram 25](#_Toc487250917)

# تعریف پروژه

## مقدمه

### هدف

این سیستم، یک نرم افزار برای رزرو زمان برای پزشکان یک بیمارستان یا ساختمان پزشکان می‌باشد. همواره یکی از دغدغه‌های پزشکان و همچنین بیماران انتخاب بهترین زمان برای بیمار بوده است. پیش از این، این وظیفه معمولا بر عهده منشی گذاشته می‌شد. که مشکلاتی ایجاد می‌کرد از قبیل :

* سخت بودن برقراری ارتباط تلفنی و همچنین مشغول شدن خطوط ساختمان پزشکان به منظور رزرو زمان بیماران
* دشواری امکان لغو یا تغییر زمان رزرو برای بیماران

لذا بر آن شدیم تا با تحلیل و طراحی یک سیستم جدید به روز، این مضل را با کمک تکنولوژی حل کنیم. با این کار، نیاز به وجود منشی برای تعیین و برنامه ریزی وقت‌های پزشک برطرف می‌شود و عملیات رزرو برای بیمار و پزشک سریعتر و آسانتر و بهینه‌تر می‌شود.

## تعریف اولیه

اتوماسیون رزرو زمان ویزیت برای بیماران و پزشکان یک ساختمان پزشکان است که این امکان را برای بیماران و همچنین پزشکان فراهم می‌کند که با صرف زمان کمتر و دقیق‌تر و بهینه‌تر زمان‌های ویزیت بیماران را ساماندهی کنند.

## تعریف کلی پروژه

در این سیستم ، پزشکان زمان‌های خالی خود را برای ویزیت بیماران اعلام می‌کنند و بیماران با مشاهده این زمان‌ها ، یک زمان را برای خود انتخاب (رزرو) می‌کنند. بیماران همچنین می‌توانند زمان انتخابی را لغو و یا آن را تغییر دهند.

علاوه براین پزشکان می‌توانند وقت‌های رزرو شده را مشاهده کنند و درصورت لزوم کل رزروهای یک روز را لغو کنند.

## نیازمندی‌ها

هر کدام از فعالیت‌ها را در قالب PIECES شرح می‌دهیم :

### بیمار

#### رزرو زمان ویزیت :

* Performance : کاربر یا همان بیمار در صورت وجود زمان خالی می‌تواند در هر زمان ممکن عمل رزرو را انجام دهد.
* Information : بیمار برای انجام رزرو باید اطلاعات روز و ساعت و تاریخ و نام دکتر را بداند.
* Economy : از هزینه‌های اضافی هم برای بیمار و هم برای پزشک جلوگیری می‌شود.
* Control : یک بیمار در یک روز نمی‌تواند دو زمان را رزرو کند و همچنین امکان رزرو باید فراهم باشد.
* Efficiency : از هزینه‌های اضافی از قبیل دستمزد منشی و استفاده بیهوده از کاغذ و همچنین اتلاف وقت بیمار جلوگیری می‌شود.
* Service : بیمار می‌تواند از این سرویس استفاده کند. نرم افزار برای بیمار باید به راحتی در دسترس باشد.

#### مشاهده جدول زمانی :

* Performance : در هر زمانی که بیمار اراده کند می‌تواند جدول را مشاهده کند.
* Information : اطلاعات روز و ساعت و تاریخ همراه با نام دکتر و قابلیت رزرو کردن در اختیار بیمار قرار بگیرد.
* Economy :
* Control : دسترسی بیمار صرفا به رزرو یا عدم رزرو یک تایم است و به جزئیات زمان رزرو شده دسترسی ندارد.
* Efficiency : نیاز به پرس و جو از منشی و اتلاف وقت بیمار از بین می‌رود.
* Service : بیمار می‌تواند از این سرویس استفاده کند. در دسترس بودن برای بیمار حائز اهمیت است.

#### تغییر زمان رزرو :

* Performance : در صورت انجام شدن عملیات رزرو، بیمار در هر زمان می‌تواند عملیات تغییر رزرو را انجام دهد.
* Information : علاوه بر اطلاعات رزرو خودش ، اطلاعات زمانهای خالی باقی مانده در اختیارش قرار بگیرد.
* Economy :
* Control : بیمار فقط اطلاعات و امکان تغییر زمان رزرو خود را دارد.
* Efficiency : نیاز به سروکله زدن با منشی نخواهد بود و همچنین از اتلاف وقت بیمار جلوگیری می‌شود.
* Service : بیمار می‌تواند از این سیستم استفاده کند. باید availabilty سیستم رعایت شود.

#### پرداخت هزینه ویزیت :

* Performance : وقتی بخواهد رزرو را انجام دهد در هر زمانی می‌تواند عملیات پرداخت را صورت دهد.
* Information : بیمار باید اطلاعات پزشک مورد نظرش و همچنین مبلغ مورد نظر را بداند.
* Economy : بیمار می‌تواند بدون پرداخت هزینه اضافی پرداخت را انجام دهد.
* Control : بیمار باید هزینه را کامل پرداخت کند و دسترسی او صرفا به قیمت و پزشک خودش است.
* Efficiency : انتقال پول از این روش هم برای بیمار و هم برای پزشک راحت‌تر و امن‌تر است.
* Service : بیمار می‌تواند از این امکان استفاده کند و باید به راحتی در اختیار بیمار قرار بگیرد.

### پزشک

#### اعلام برنامه :

* Performance : پزشکان باید اول هر ماه برنامه شان را اعلام کنند.
* Information : تقویم هفته باید در اختیار پزشک قرار بگیرد.
* Economy : برای پزشک واسطه منشی حذف می‌شود و مشکلاتی مثل هزینه اضافی منشی حذف خواهد شد.
* Control : پزشک در روزهای تعطیل نمی‌تواند برنامه داشته باشد.
* Efficiency : پزشک برنامه کاری خودش را دقیقتر و بهتر تنظیم می‌کند.
* Service : پزشک از این سیستم اسفتاده می‌کند. سیستم قابل دسترس باشد.

#### مشاهده زمان‌های رزرو شده‌ی بیماران :

* Performance : بدون هیچ محدودیتی در هر زمان می‌تواند زمانهای رزرو شده را مشاهده کند.
* Information : تمامی اطلاعات تایم رزرو شده و همهی اطلاعات بیمار در اختیار پزشک قرار می‌گیرد.
* Economy :
* Control : هر پزشک فقط می‌تواند اطلاعات بیماران خودش را داشته باشد.
* Efficiency : پزشک بهتر می‌تواند زمان و برنامه‌اش را تنظیم کند.
* Service : پزشک از این سیستم اسفتاده می‌کند. سیستم قابل دسترس باشد.

#### لغو کردن زمان‌های رزرو شده :

* Performance : پزشک نهایتا سه روز قبل از روز لغو ، فرآیند لغو را انجام دهد.
* Information : زمان‌های رزرو شده توسط بیماران را ببیند.
* Economy :
* Control : پزشک فقط می‌تواند برنامه یک روز کامل را لغو کند.
* Efficiency : بیمار و همچنین پزشک می‌توانند زمانشان را تنظیم و برنامه ریزی کنند.
* Service : پزشک از این سیستم اسفتاده می‌کند. سیستم قابل دسترس باشد.

### مدیر سیستم

#### مشاهده زمان‌های رزرو :

* Performance : مدیر سیستم در هر زمانی می‌تواند به مشاهده زمان‌های رزرو دسترسی داشته باشد.
* Information : مدیر سیستم باید اطلاعات بیمار و همچنین پزشک را در اختیار داشته باشد.
* Economy :
* Control : دسترسی‌ای به تغییر اطلاعات کاربران ندارد.
* Efficiency : مدیر از اتفاقاتی که در سیستم می‌افتد باخبر است و همچنین از بی‌نظمی‌های احتمالی با مدیریت و مشاهده اعمال توسط مدیر جلوگیری می‌شود.
* Service : مدیر سیستم از این سیستم اسفتاده می‌کند. باید قابل دسترس باشد.

#### ثبت نام بیمار جدید :

* Performance : هر زمان که یک بیمار درخواست ثبت نام داده باشد.
* Information : باید اطلاعات بیمار را برای ثبت نام در اختیار داشته باشد که به سیستم بدهد.
* Economy :
* Control : فقط به اطلاعات بیمار مورد نظر دسترسی داشته باشد.
* Efficiency : با دسترسی مدیر برای ثبت نام از وارد شدن داده‌های حشو به سیستم جلوگیری می‌شود.
* Service : مدیر سیستم از این سیستم استفاده می‌کند و باید قابل دسترس باشد.

#### ثبت نام دکتر جدید :

* Performance : هر زمان که یک پزشک درخواست ثبت نام داده باشد.
* Information : باید اطلاعات پزشک را برای ثبت نام در اختیار داشته باشد که به سیستم بدهد.
* Economy :
* Control : فقط به اطلاعات پزشک مورد نظر دسترسی داشته باشد.
* Efficiency : با دسترسی مدیر برای ثبت نام از وارد شدن داده‌های حشو به سیستم جلوگیری می‌شود.
* Service : مدیر سیستم از این سیستم استفاده می‌کند و باید قابل دسترس باشد.

### فاعل زمان

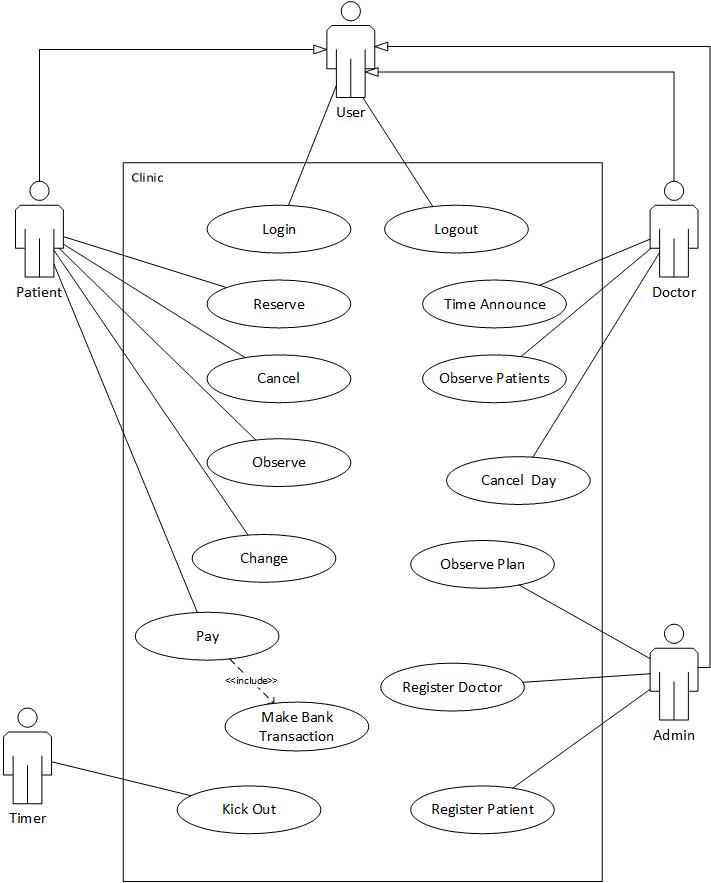
#### خارج کردن کاربر از سیستم :

* Performance : بعد از 15 دقیقه که کاربر کاری با سیستم ندارد او را از سیستم خارج کند.
* Information : صرفا باید بداند که کدام کاربر از چه زمانی در سیستم وارد شده است.
* Economy :
* Control : به اطلاعات شخصی بیمار یا پزشک دسترسی ندارد.
* Efficiency : از شلوغ شدن سیستم و پایین آمدن کارآمدی‌اش کاسته می‌شود.
* Service : فاعل زمان به این سیستم دسترسی دارد.

# Structured Analysis

## Use Case

### Use Case Diagram



## Use Case Glossary

|  |  |  |
| --- | --- | --- |
| Reserve | | Use case name |
| 01 | | Use case id |
| Medium | | Priority |
| گزارش نیازمندی ها | | Source |
| بیماران | | Primary  Business actor |
| پزشکان | | Other  Participate  actors |
|  | | Other interested  stakeholders |
| در این use case بیماران زمان مورد نظر خود را از پزشک مورد نظر رزرو میکنند | | Description |
| بیمار باید در سیستم ثبت نام کرده باشد | | Precondition |
| این use case زمانی فعال میشود که بیمار در خواست رزرو داشته باشد | | Trigger |
| Actor Action | System Response | Typical course of  events |
| گام ۱ : بیمار درخواست رزور میدهد  گام ۳ : بیمار از بین تایم های خالی یکی را انتخاب میکند | گام ۲ : سیستم با توجه به پزشک مورد نظر تایم های خالی را ارایه میدهد  گام ۴ : در صورت موفق بودن تایم مورد نظر رزرو میشود و پیام موفقیت به بیمار داده میشود |
|  | | Alternate course |
| زمان مورد نظر بیمار رزرو میشود | | Conclusion |
|  | | Postcondition |
|  | | Business rules |
| لیستی از پزشکان وجود دارد و گزینه ای برای تایید که با فشردن آن به بیمار لیست پزشک مود نظر او نشان داده میشود. | | Implementation  Constraints and  specification |
|  | | Assumptions |
|  | | Open issues |

|  |  |  |
| --- | --- | --- |
| Cancel | | Use case name |
| 02 | | Use case id |
| Medium | | Priority |
| گزارش نیازمندی ها | | Source |
| بیماران | | Primary  Business actor |
| پزشک | | Other  Participate  actors |
|  | | Other interested  stakeholders |
| بیمار تایم قبلی ای را که رزور کرده بود، کنسل میکند | | Description |
| این عمل تنها تا ۴۸ ساعت قبل از وقت بیمار ممکن است و همچنین بیمار باید در تایمی را از قبل رزرو کرده باشد | | Precondition |
| زمانی فعال میشود که بیمار بخواهد زمانی را لغو کند | | Trigger |
| Actor Action | System Response | Typical course of  events |
| گام ۱ : بیمار درخواست لغو ویزیت را میکند | گام ۲ : سیستم بررسی میکند آیا در بازه زمانی لغو ویزیت هستیم یا خیر؟  گام ۳ : در صورت موفق بودن گام قبل ویزیت لغو میشود  گام ۴ : پیام مناسب با وضعیت درخواست به بیمار نشان داده میشود. |
| در صورت موفق بودن عمل لغو، بیمار میتواند زمان دیگری را رزرو کند. | | Alternate course |
| زمان مورد نظر بیمار لغو میشود. | | Conclusion |
|  | | Postcondition |
|  | | Business rules |
| در قسمت زمان های رزور شده، کنار هر زمان گزینه ای برای در قسمت زمان های رزور شده، کنار هر زمان گزینه ای برای لغو آن زمان وجود دارد که بیمار با فشار دادن آن میتواند زمان مشخصی را لغو کند. | | Implementation  Constraints and  specification |
|  | | Assumptions |
|  | | Open issues |

|  |  |  |
| --- | --- | --- |
| Change reservation | | Use case name |
| 03 | | Use case id |
| Medium | | Priority |
| گزارش نیازمندی ها | | Source |
| بیمار | | Primary  Business actor |
| پزشک | | Other  Participate  actors |
|  | | Other interested  stakeholders |
| بیمار میتواند زمان ویزیت را تغییر دهد | | Description |
| این عمل تنها تا ۴۸ ساعت قبل از وقت بیمار ممکن است و همچنین بیمار باید در تایمی را از قبل رزرو کرده باشد | | Precondition |
| زمانی فعال میشود که بیمار بخواهد زمان ویزیت را تغییر بدهد. | | Trigger |
| Actor Action | System Response | Typical course of  events |
| گام ۱ : بیمار درخواست تغییر زمان ویزیت را میکند  گام ۵ : بیمار تایم جدید مورد نظر را انتخاب میکند | گام ۲ : سیستم بررسی میکند آیا در بازه زمانی تغییر زمان ویزیت هستیم یا خیر؟  گام ۳ : در صورت موفق بودن گام قبل ویزیت لغو میشود  گام ۴ : تایم های خالی پزشک مورد نظر مجددا به بیمار نشان داده میشود.  گام ۶ : پیام مناسب با وضعیت درخواست به بیمار نشان داده میشود. |
|  | | Alternate course |
| زمان ویزیت بیمار تغییر میکند. | | Conclusion |
|  | | Postcondition |
|  | | Business rules |
| در قسمت زمان های رزور شده، کنار هر زمان گزینه ای برای تغییر زمان ویزیت وجود دارد که بیمار با فشردن آن میتواند زمان های خالی پزشک مورد نظر ببیند و دوباره زمان جدیدی را برای ویزیت انتخاب کند. | | Implementation  Constraints and  specification |
|  | | Assumptions |
|  | | Open issues |

|  |  |  |
| --- | --- | --- |
| Observe times | | Use case name |
| 04 | | Use case id |
| Medium | | Priority |
| گزارش نیازمندی ها | | Source |
| بیمار | | Primary  Business actor |
| پزشک | | Other  Participate  actors |
|  | | Other interested  stakeholders |
| بیمار میتواند زمان های خالی پزشک مورد نظرش را ببیند. | | Description |
|  | | Precondition |
| زمانی فعال میشود که پزشک بخواهد زمان هایی را که بیماران با او رزرو کرده اند را ببیند. | | Trigger |
| Actor Action | System Response | Typical course of  events |
| گام ۱ : بیمار پزشک مورد نظر را انتخاب میکند. | گام ۲ : سیستم زمان های خالی پزشک را به بیمار نشان میدهد. |
|  | | Alternate course |
| زمان های ویزیت پزشک را به او نشان میدهند. | | Conclusion |
|  | | Postcondition |
|  | | Business rules |
| گزینه ای در میزکاربری پزشک وجود دارد که با فشردن آن و مشخص کردن روز مورد نظر میتواند زمان های رزرو شده در آن روز را مشاهده کند. | | Implementation  Constraints and  specification |
|  | | Assumptions |
|  | | Open issues |

|  |  |  |
| --- | --- | --- |
| Time announce | | Use case name |
| 06 | | Use case id |
| Low | | Priority |
| گزارش نیازمندی ها | | Source |
| پزشک | | Primary  Business actor |
|  | | Other  Participate  actors |
|  | | Other interested  stakeholders |
| پزشک زمان هایی را که میتواند بیمار ویزیت کند را اعلام میکند | | Description |
|  | | Precondition |
| زمانی فعال میشود که پزشک بخواهد زمان های مورد نظر برای پذیرفتن بیمار را مشخص کند. | | Trigger |
| Actor Action | System Response | Typical course of  events |
| گام ۱ : پزشک در خواست اعلام زمان میکند  گام ۳ : پزشک تغییرات مورد نظر را اعمال میکند | گام ۲ : سیستم زمان های مشخص کرده او را به همراه امکان تغییر و ایجاد زمان جدید به او نشان میدهد  گام ۴ : سیستم تغییرات مورد نظر پزشک را ثبت میکند. |
|  | | Alternate course |
| زمان های مورد نظر پزشک برای پذیرفتن بیمار مشخص میشود. | | Conclusion |
|  | | Postcondition |
|  | | Business rules |
| پزشک با فشردن گزینه ای میتواند جدول روز ها را مشاهده کند و با مشخص کردن روز مورد نظر، جدول زمان را مشاهده میکند که میتواند تایم های مورد نظر را در آن روز مشخص کند. | | Implementation  Constraints and  specification |
|  | | Assumptions |
|  | | Open issues |

|  |  |  |
| --- | --- | --- |
| Observe patients | | Use case name |
| 07 | | Use case id |
| Medium | | Priority |
| گزارش نیازمندی ها | | Source |
| پزشک | | Primary  Business actor |
| بیمار | | Other  Participate  actors |
|  | | Other interested  stakeholders |
| پزشک زمان هایی را که بیماران رزرو کرده اند را میبیند | | Description |
|  | | Precondition |
| زمانی فعال میشود که پزشک بخواهد زمان های رزرو شده خودش را مشاهده کند. | | Trigger |
| Actor Action | System Response | Typical course of  events |
| گام ۱ : پزشک در خواست اعلام زمان های رزرو شده را میکند | گام ۲ : سیستم زمان های رزرو شده را به پزشک نشان میدهد |
|  | | Alternate course |
|  | | Conclusion |
|  | | Postcondition |
|  | | Business rules |
| گزینه ای در میز کاربری پزشک وجود دارد که با فشردن آن و سپس انتخاب روز مورد نظر میتواند زمان هایی که بیمار به او مراجعه میکند را مشاهده کند. | | Implementation  Constraints and  specification |
|  | | Assumptions |
|  | | Open issues |

|  |  |  |
| --- | --- | --- |
| Cancel a day | | Use case name |
| 08 | | Use case id |
| High | | Priority |
| گزارش نیازمندی ها | | Source |
| پزشک | | Primary  Business actor |
| بیمار | | Other  Participate  actors |
|  | | Other interested  stakeholders |
| پزشک میتواند یک روز را به طور کامل کنسل کند | | Description |
|  | | Precondition |
| زمانی فعال میشود که پزشک بخواهد روزی را کنسل کند. | | Trigger |
| Actor Action | System Response | Typical course of  events |
| گام ۱: پزشک در خواست کنسل کردن روز را میکند | گام ۲: سیستم روز مورد نظر او را کنسل میکند  گام ۳: سیستم پیغام مورد نیاز را به پزشک میدهد |
|  | | Alternate course |
|  | | Conclusion |
|  | | Postcondition |
|  | | Business rules |
| پزشک میتواند در جدول زمان هایی که به او مراجعه میشود با فشردن گزینه ای آن زمان مورد نظر را کنسل کند. | | Implementation  Constraints and  specification |
|  | | Assumptions |
|  | | Open issues |

|  |  |  |
| --- | --- | --- |
| Login | | Use case name |
| 09 | | Use case id |
| High | | Priority |
| گزارش نیازمندی ها | | Source |
| کاربران | | Primary  Business actor |
|  | | Other  Participate  actors |
|  | | Other interested  stakeholders |
| در این use case هر کاربر باید نام کاربری و رمز عبور خود را وارد کند و با توجه به این دو سیستم تصمیم میگیرد که کاربر سفارش دهنده غذا است یا مسئول آشپزخانه و پنل مربوط به هر کدام را برای کاربر نمایش می دهد. | | Description |
|  | | Precondition |
| این use case زمانی فعال میشود که کاربر درخواست ورود به سیستم را داشته باشد. | | Trigger |
| Actor Action | System Response | Typical course of  events |
| گام ۱ : کاربر نام کاربری و کلمه عبور خود را وارد میکند و دکمه ورود را می زند. | گام ۲ : سیستم نام کاربری و کلمه عبور را بررسی میکند و در صورت صحت اجاره ورود میدهد.  گام ۳ : با توجه به نام کاربری تشخیص داده که کاربر پزشک است یا بیمار و سپس پنل مورد نظر را به کاربر نمایش می دهد |
| گام ۱ جایگزین : در صورت دریافت خطا کاربر دوباره نام کاربری و کلمه عبور صحیح خود را وارد میکند.  گام ۳ جایگزین : اگر نام کاربری و یا کلمه عبور اشتباه بود پیام خطا نمایش می دهد. | | Alternate course |
| کاربر با موفقیت وارد سامانه می شود | | Conclusion |
|  | | Postcondition |
|  | | Business rules |
| باید یک صفحه برای ورود شامل دو فیلد برای وارد کردن نام کاربری و کلمه عبور و یک دکمه ورود باشد و اگر نام کاربری یا کلمه عبور اشتباه بود اجازه دسترسی داده نشود. | | Implementation  Constraints and  specification |
|  | | Assumptions |
|  | | Open issues |

|  |  |  |
| --- | --- | --- |
| Logout | | Use case name |
| 10 | | Use case id |
| High | | Priority |
| گزارش نیازمندی ها | | Source |
| کاربران | | Primary  Business actor |
|  | | Other  Participate  actors |
|  | | Other interested  stakeholders |
| در این use case هر کاربر با زدن دکمه خروج از سیستم خارج میشود و برای استفاده مجدد باید دوباره وارد شود. | | Description |
| هر کاربر باید از قبل وارد سیستم شده باشد. | | Precondition |
| این use case وقتی هر کسی اقدام به خروج از پنل کاربری خود کند فعال میشود. | | Trigger |
| Actor Action | System Response | Typical course of  events |
| گام ۱ : کاربر دکمه خروج را می زند. | گام ۲ : سیستم کاربر را خارج میکند و برای استفاده مجدد از سیستم, کاربر باید وارد شود. |
|  | | Alternate course |
| کاربر با موفقیت از سامانه خارج می شود | | Conclusion |
| کاربر از سیستم خارج شده و صفحه ورود برای وی نمایش داده میشود. | | Postcondition |
|  | | Business rules |
| سیستم یک دکمه خروج دارد. | | Implementation  Constraints and  specification |
| کاربران از قبل وارد سیستم شده اند. | | Assumptions |
|  | | Open issues |

|  |  |  |
| --- | --- | --- |
| Kick out | | Use case name |
| 14 | | Use case id |
| Medium | | Priority |
| گزارش نیازمندی ها | | Source |
| زمان | | Primary  Business actor |
|  | | Other  Participate  actors |
|  | | Other interested  stakeholders |
| در این usecase باید هر کاربری که مدت زمان ورود آن منقضی شده است از سیستم خارج شود. | | Description |
| کاربر باید ابتدا login کرده باشد. | | Precondition |
| هر ۱۲ ساعت یکبار باید اجرا گردد | | Trigger |
| Actor Action | System Response | Typical course of  events |
| گام ۱ : در زمان مقرر سیستم اقدام به خارج کردن کاربران میکند. | گام ۲ : سیستم کاربر را خارج میکند و برای استفاده مجدد از سیستم, کاربر باید وارد شود. |
|  | | Alternate course |
| کاربر با موفقیت از سامانه خارج می شود | | Conclusion |
| کاربر از سیستم خارج شده و صفحه ورود برای وی نمایش داده میشود. | | Postcondition |
|  | | Business rules |
| به صورت یک background service باید پیاده سازی گردد. | | Implementation  Constraints and  specification |
| کاربران از قبل وارد سیستم شده اند. | | Assumptions |
|  | | Open issues |

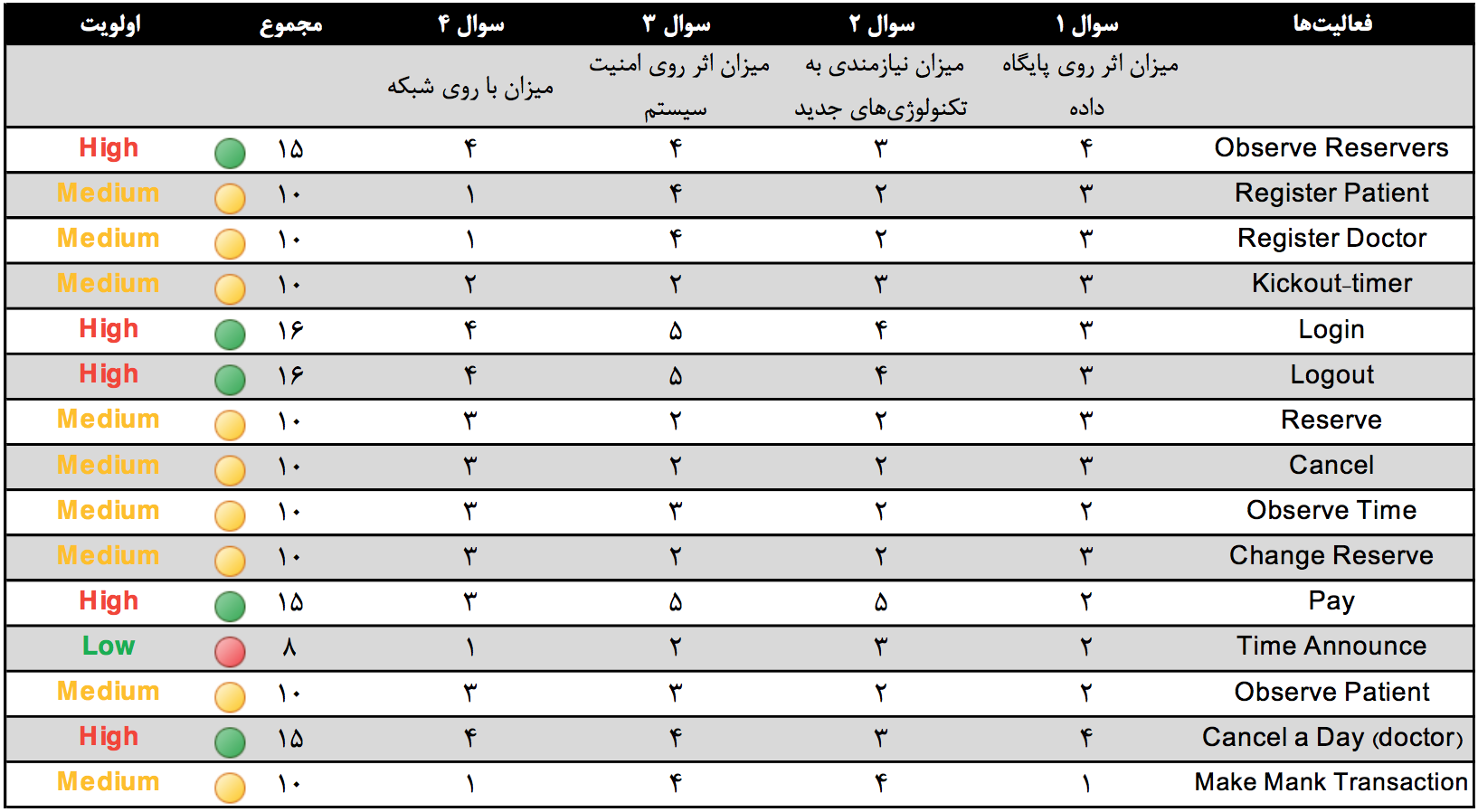
|  |  |  |
| --- | --- | --- |
| pay | | Use case name |
| 05 | | Use case id |
| High | | Priority |
| گزارش نیازمندی ها | | Source |
| پزشک | | Primary  Business actor |
|  | | Other  Participate  actors |
|  | | Other interested  stakeholders |
| اتصال به درگاه بانکی برای انتقال اعتبار برقرار میشود. | | Description |
| کاربر باید عضو شبکه شتاب باشد. | | Precondition |
| این use case وقتی که کاربر برای پرداخت پول سعی دارد به درگاه بانکی متصل شود فعال میشود. | | Trigger |
| Actor Action | System Response | Typical course of  events |
| گام ۱ : دانشجو برای پرداخت اقدام میکند. | گام ۲ : سیستم با درخواستی به درگاه بانکی متصل میشود.  گام 3: به سایت برمیگردد و با توجه به موفقیت و عدم موفقیت عملیات، پیام مناسب را نمایش میدهد. |
|  | | Alternate course |
| با موفقیت به درگاه بانک عضو شتاب متصل میشویم. | | Conclusion |
|  | | Postcondition |
|  | | Business rules |
| با ارسال درخواستی به API بانک کاربر را به سایت بانک منتقل میکنیم. | | Implementation  Constraints and  specification |
|  | | Assumptions |
|  | | Open issues |

|  |  |  |
| --- | --- | --- |
| Observe reserves | | Use case name |
| 11 | | Use case id |
| High | | Priority |
| گزارش نیازمندی ها | | Source |
| ادمین | | Primary  Business actor |
|  | | Other  Participate  actors |
|  | | Other interested  stakeholders |
| ادمین میتواند زمان های رزرو شده تمام پزشکان را مشاهده کند. | | Description |
|  | | Precondition |
| این use case وقتی که کاربر برای پرداخت پول سعی دارد به درگاه بانکی متصل شود فعال میشود. | | Trigger |
| Actor Action | System Response | Typical course of  events |
| گام ۱ : ادمین پزشک مورد نظر را انتخاب میکند. | گام ۲ : سیستم جدول رزور پزشک مورد نظر را نشان میدهد. |
|  | | Alternate course |
| ادمین با موفقیت جدول پزشک مورد نظر را میبیند. | | Conclusion |
|  | | Postcondition |
|  | | Business rules |
| یک لیست از پزشک ها وجود دارد که ادمین با انتخاب پزشک مورد نظر و فشردن گزینه تایید، جدول ایشان را مشاهده میکند. | | Implementation  Constraints and  specification |
|  | | Assumptions |
|  | | Open issues |

|  |  |  |
| --- | --- | --- |
| Register Doctor | | Use case name |
| 12 | | Use case id |
| Medium | | Priority |
| گزارش نیازمندی ها | | Source |
| ادمین سیستم | | Primary  Business actor |
|  | | Other  Participate  actors |
|  | | Other interested  stakeholders |
| در این use case ادمین اقدام به تایید ثبت نام پزشکان سامانه میکند. با این اطلاعات پزشکان میتوانند وارد سامانه شوند. | | Description |
| هرگاه پزشک جدیدی وارد سیستم شود, ادمین اقدام به تایید ثبت نام وی میکند. اگر ثبت نام انجام نشود ورود پزشک ممکن نیست. | | Precondition |
| این use case وقتی که ادمین ثبت نام پزشک را تایید کند فعالسازی میشود. | | Trigger |
| Actor Action | System Response | Typical course of  events |
| گام ۱ : ادمین اطلاعات پزشکان جدید را تایید میکند. | گام ۲ : سیستم اطلاعات ثبت نامی را ارزیابی میکند و در صورت عدم وجود مشکل (اطلاعات درست) اطلاعات در پایگاه داده ثبت میشود. |
|  | | Alternate course |
| اطلاعات پزشکان در سامانه ثبت میشود و برای ورود مورد استفاده قرار میگیرد. | | Conclusion |
|  | | Postcondition |
|  | | Business rules |
| باید یک فرم اطلاعات وارد شده توسط بیمار برای ادمین نشان داده شود تا بتواند اطلاعات پزشکان جدید را تایید کند و اقدام به ثبت نام آنها کند. | | Implementation  Constraints and  specification |
|  | | Assumptions |
|  | | Open issues |

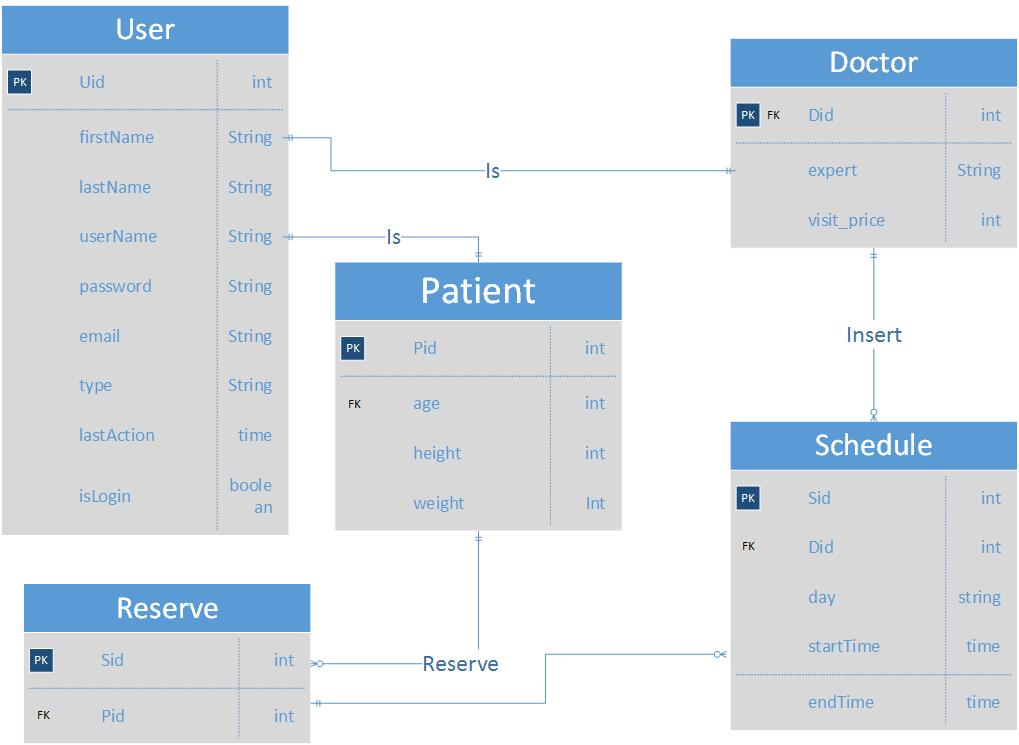
|  |  |  |
| --- | --- | --- |
| Register Patient | | Use case name |
| 13 | | Use case id |
| Medium | | Priority |
| گزارش نیازمندی ها | | Source |
| ادمین سیستم | | Primary  Business actor |
|  | | Other  Participate  actors |
|  | | Other interested  stakeholders |
| در این use case ادمین اقدام به تایید ثبت نام بیماران سامانه میکند. با این اطلاعات بیماران میتوانند وارد سامانه شوند. | | Description |
| هرگاه بیمار جدیدی وارد سیستم شود, ادمین اقدام به تایید ثبت نام وی میکند. اگر ثبت نام انجام نشود ورود بیمار ممکن نیست. | | Precondition |
| این use case وقتی که ادمین ثبت نام بیماران را تایید کند فعالسازی میشود. | | Trigger |
| Actor Action | System Response | Typical course of  events |
| گام ۱ : ادمین اطلاعات بیماران جدید را تایید میکند. | گام ۲ : سیستم اطلاعات ثبت نامی را ارزیابی میکند و در صورت عدم وجود مشکل (اطلاعات درست) اطلاعات در پایگاه داده ثبت میشود. |
|  | | Alternate course |
| اطلاعات بیماران در سامانه ثبت میشود و برای ورود مورد استفاده قرار میگیرد. | | Conclusion |
|  | | Postcondition |
|  | | Business rules |
| باید یک فرم اطلاعات وارد شده توسط بیمار برای ادمین نشان داده شود تا بتواند اطلاعات بیمار جدید را تایید کند و اقدام به ثبت نام آنها کند. | | Implementation  Constraints and  specification |
|  | | Assumptions |
|  | | Open issues |

### Priority Table

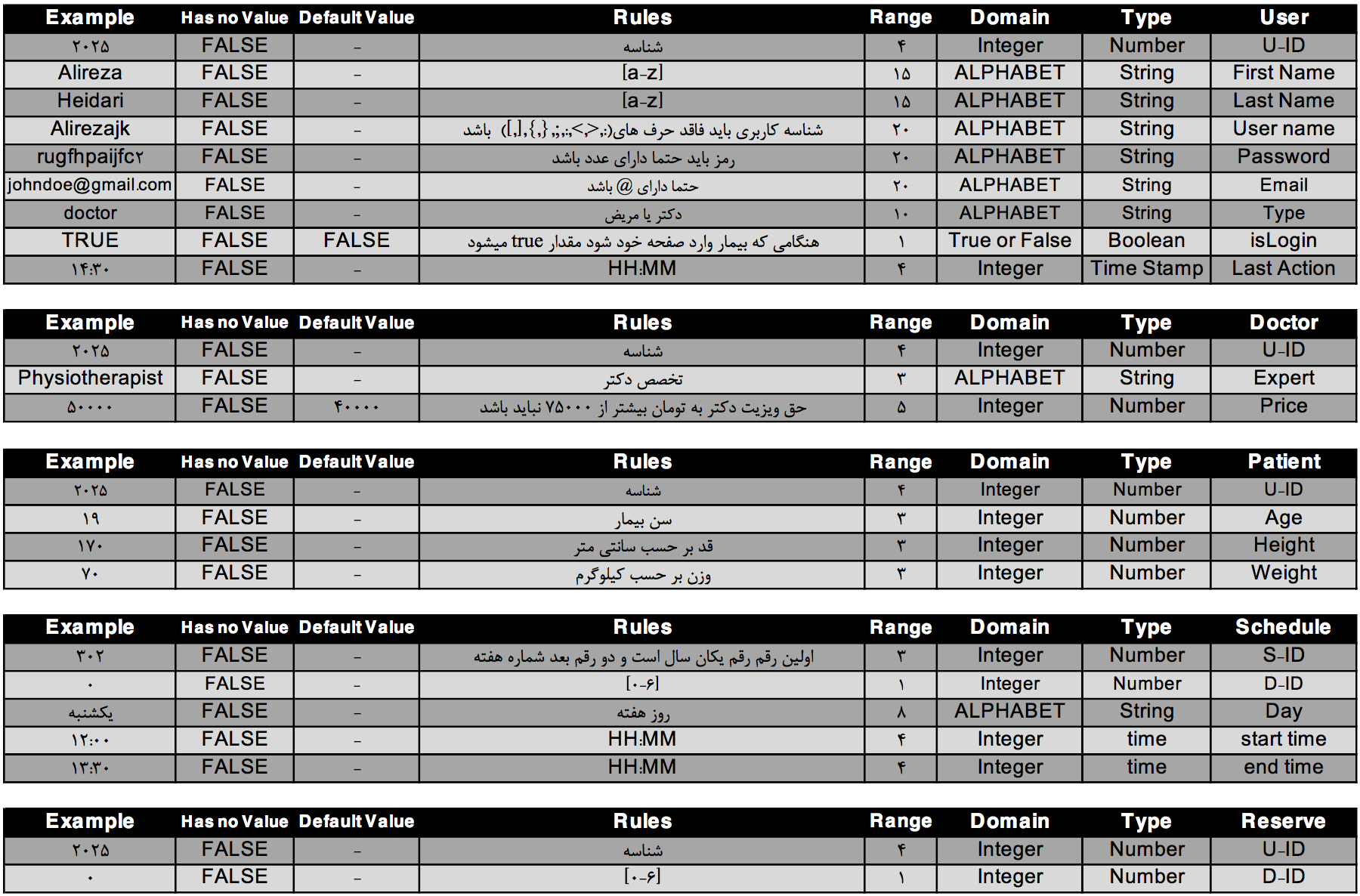


## Data Model

### Entity Relationship Diagram

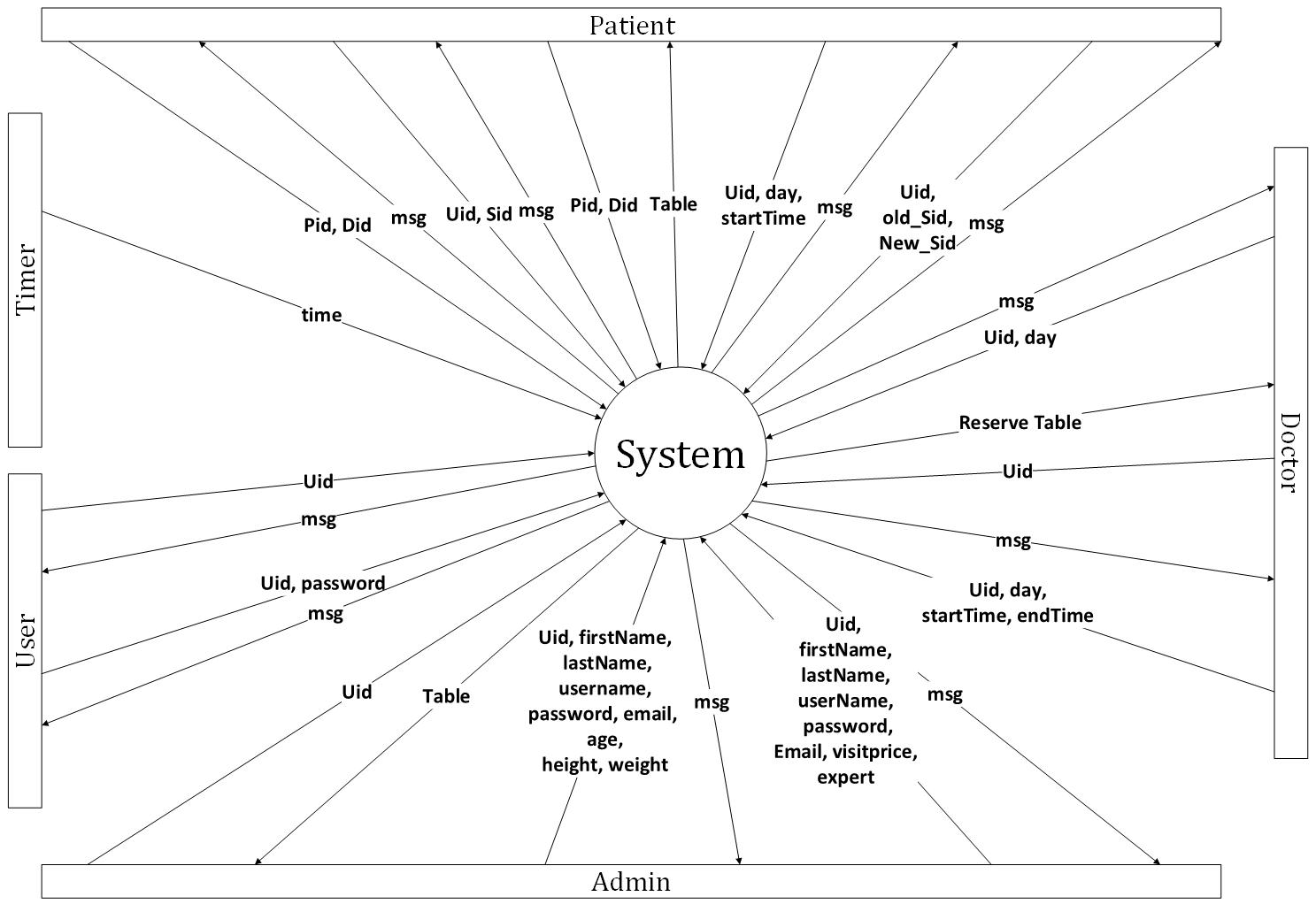


### Object Descriptor

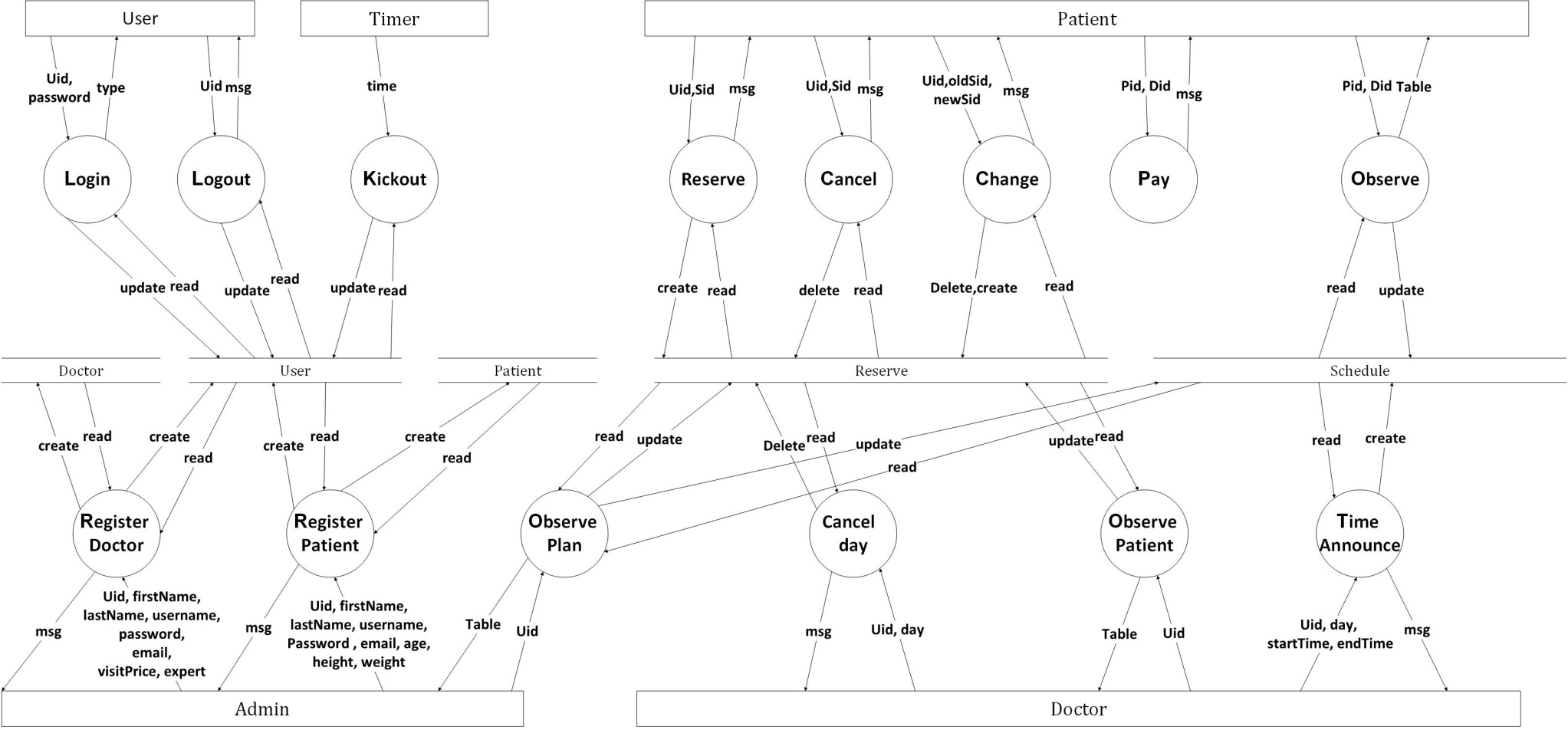


## Functional Model

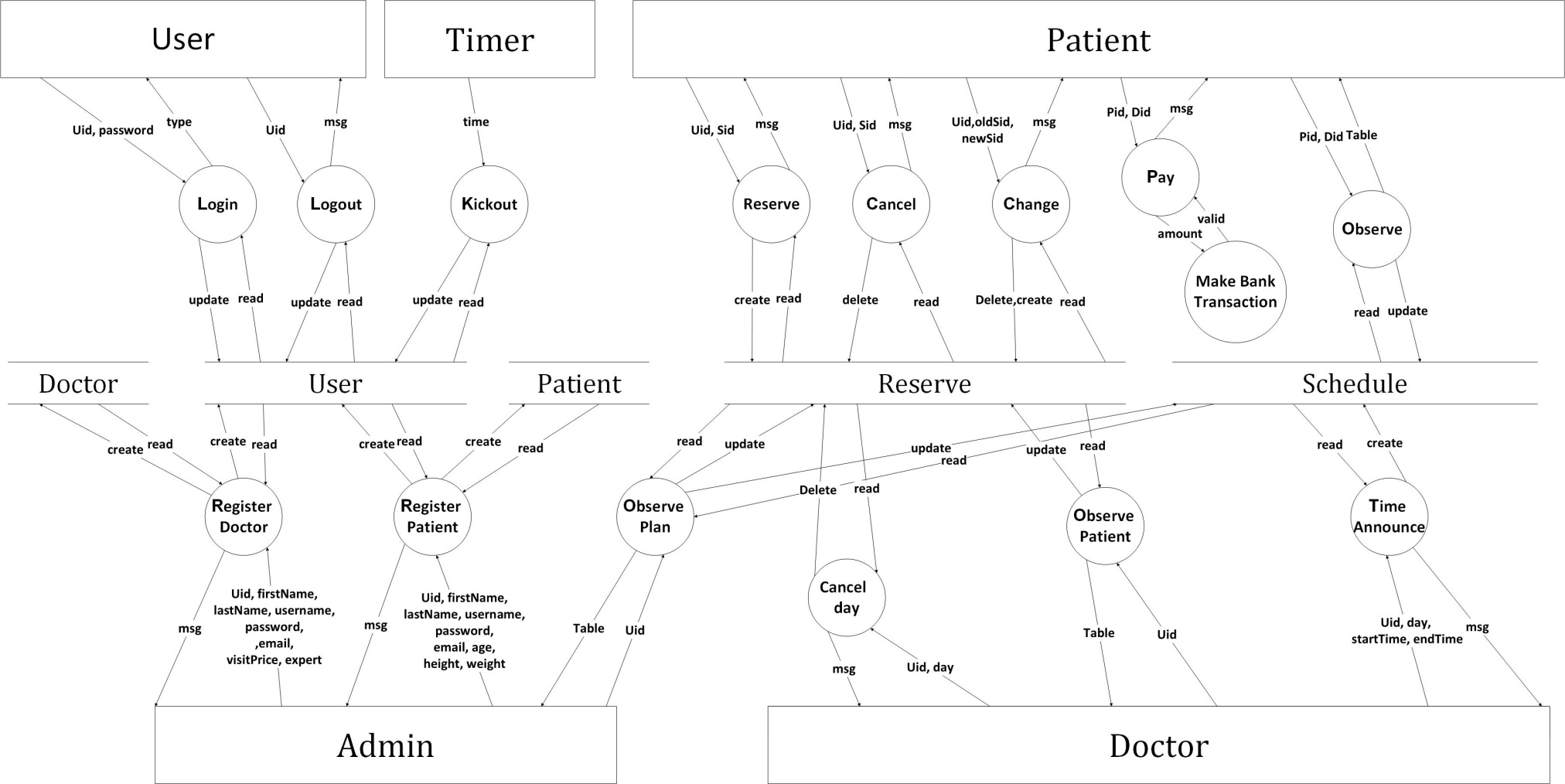
### Data Flow Diagram Level 0



### Data Flow Diagram Level 1



### Data Flow Diagram Level 2



### Process Specification

Login (Uid, password) {

u = UserFind (Uid);

valid = (u! = null);

if (!valid) {

return("Your account doesn't exist");

} else {

valid2 = (u.getPassword() == password);

if (valid2) {

u.setIsLogin(true);

u.setLastAction (getCurrentTime());

type = UserGetUserType(Uid);

return type;

} else {

return("your password is incorrect");

}

}

}

Logout(Uid) {

u = UserFind(Uid);

u.setIsLogin(false);

return("successful!");

}

Kickout (time) {

UsersVector = UserFindAll();

for(int i = 0 ; i < UsersVector.size(); i++){

u = UsersVector.get(i);

status = (time - u.getLastAction() > 15 min);

if (status == true) {

u.setIsLogin(false);

}

}

Reserve (Uid, Sid) {

u = UserFind (Uid);

u.setLastAction (getCurrentTime());

s = ReserveFind(Sid);

valid = (s == null);

if (valid) {

reserveStatus = ReserveAdd(Uid,Sid);

if(reserveStatus)

return ("Successful");

else

return ("Unsuccessful");

} else

return ("This time is full!");

}

Cancel (Uid, Sid) {

u = UserFind(Uid);

u.setLastAction(getCurrentTime());

scheduleRecord = ScheduleFind(Sid);

time = (scheduleRecord.getDay()- getCurrentTime() > 1 Day);

if (time == true) {

result = ReserveDelete(Sid);

if (result == true)

return ("Cancelation Successful");

else

return ("Unsuccessful");

} else {

return ("Request time out !");

}

Change (Uid, oldSid, newSid) {

u = UserFind(Uid);

u.setLastAction(getCurrentTime());

r = ReserveFind(oldSid);

r.setSid(newSid);

return("Successful");

}

Pay (Pid, Did) {

u = UserFind(Pid);

u.setLastAction(getCurrentTime());

price = DoctorGetVisitPrice(Did);

bankBool = makeBankTransaction(price);

if (bankBool == true)

return (“Successful”);

else

return (“Unsuccessful”);

}

makeBankTransaction (amount) {

transacted = transact(amount);

valid = (amount == transacted);

return valid;

}

Observe (Pid, Did){

u = UserFind(Pid);

u.setLastAction(getCurrentTime());

scheduleTable = ScheduleShowTableByDoctorId(Did);

valid = (scheduleTable == null);

if(valid)

return ("The doctor doesn't exist");

else

return scheduleTable;

}

}

RegisterDoctor (Uid, firstName, lastName, username, password,

email, visitPrice, expert) {

u = UserFindAdmin();

u.setLastAction(getCurrentTime());

result = UserInsert(Uid, firstName,

lastName, username,

password, email,

"doctor", true,

getCurrentDateTime);

if (result == true){

doctorStatus =

DoctorInsert(Uid, expert, visitPrice);

if (doctorStatus == true)

return (“Added Successfully”);

else

return (“Unsuccessful”);

}else

return (“User has already exists”);

}

RegisterPatient (Uid, firstName, LastName, username, password,

email, age , height , weight){

u = UserFindAdmin();

u.setLastAction(getCurrentTime());

result = UserInsert(Uid, firstName, lastName ,

username, password, email ,

"patient" , true ,

getCurrentTime());

if(result == false)

return ("this user had been added before");

else{

patientStatus =

PatientInsert(Uid, age, height , weight);

if(patientStatus)

return ("Successful");

else

return ("Unsuccessful");

}

}

ObservePlan (Uid) {

u = UserFind(Uid);

u.setLastAction(getCurrentTime());

reserveTable = getReserveTable();

return reserveTable;

}

CancelDay (Uid, day) {

u = UserFind(Uid);

u.setLastAction(getCurrentTime());

time = (getCurrentTime() – day >= 3 day);

if(time == true) {

patientsVector = ReserveFindAllPatient(Uid);

sendEmailTo(patientsVector);

reserveStatus = ReserveDelete(Uid);

scheduleStatus = ScheduleDelete(Uid,day);

if(reserveSatatus && scheduleStatus)

return ("Successful");

else

return ("Unsuccessful");

}

else

return ("You can't cancel this day");

}

ObservePatient (Uid) {

u = UserFind(Uid);

u.setLastAction(getCurrentTime());

reserveTable = getReserveTableByDoctorID(Uid);

return reserveTable;

}

TimeAnnounce (Uid, day, startTime, endTime) {

u = UserFind(Uid);

u.setLastAction(getCurrentTime());

scheduleStatus =

ScheduleInsert(Uid, date, startTime, endTime);

if(scheduleStatus == true)

return ("Successful");

else

return ("Unsuccessful");

}

}

## Behavioral Model

### State Transitoin Diagram

