

MEET WITH

1401
fall semester
Dr M.Bejani



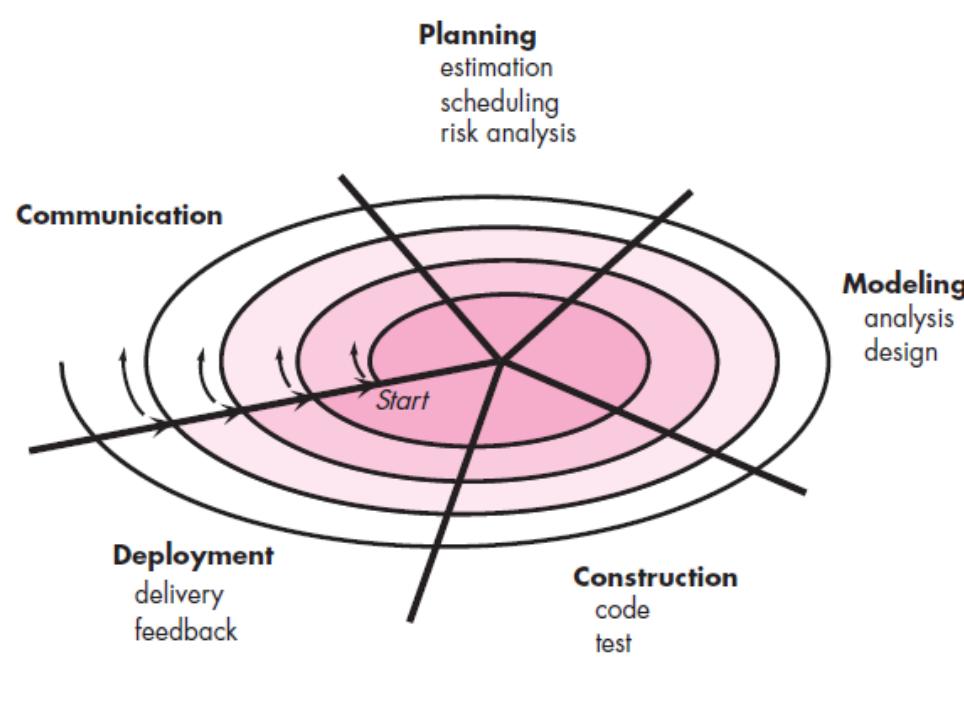
HELLO!

We are
MohammadReza Bateni,
Abolfazl ArabAnsari,
Ali Mohammadpour,
Zahra Nekounam
and
Negin Rahimi



2.

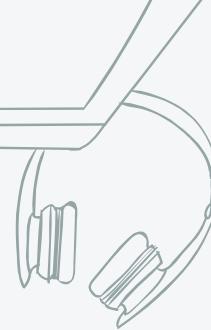
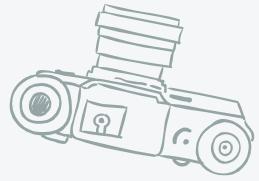
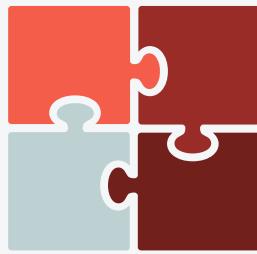
PROCESS MODEL



Spiral Process Model

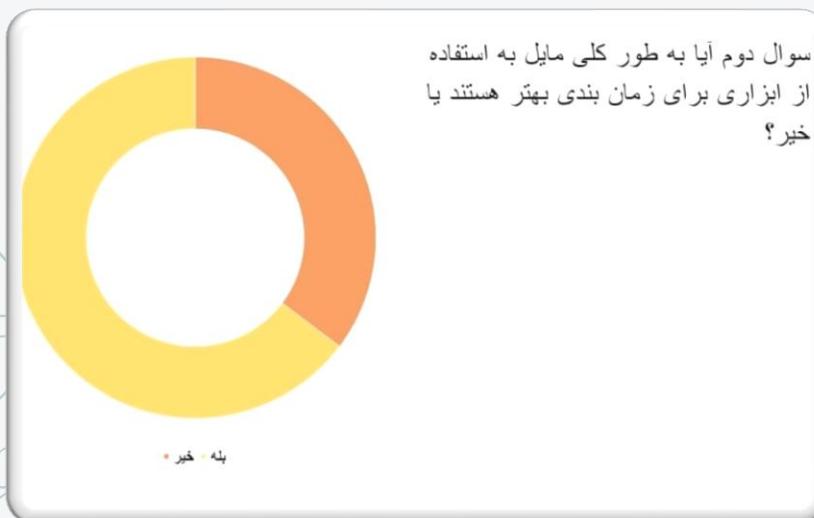
Building a Web App

6



NEEDS ASSESSMENT

of professors

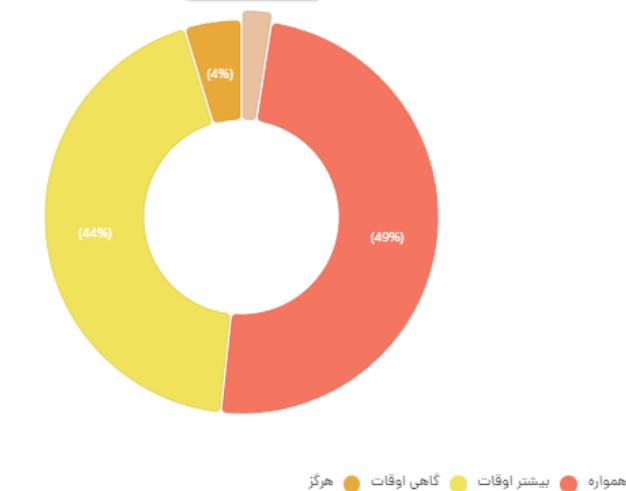


NEEDS ASSESSMENT

of students

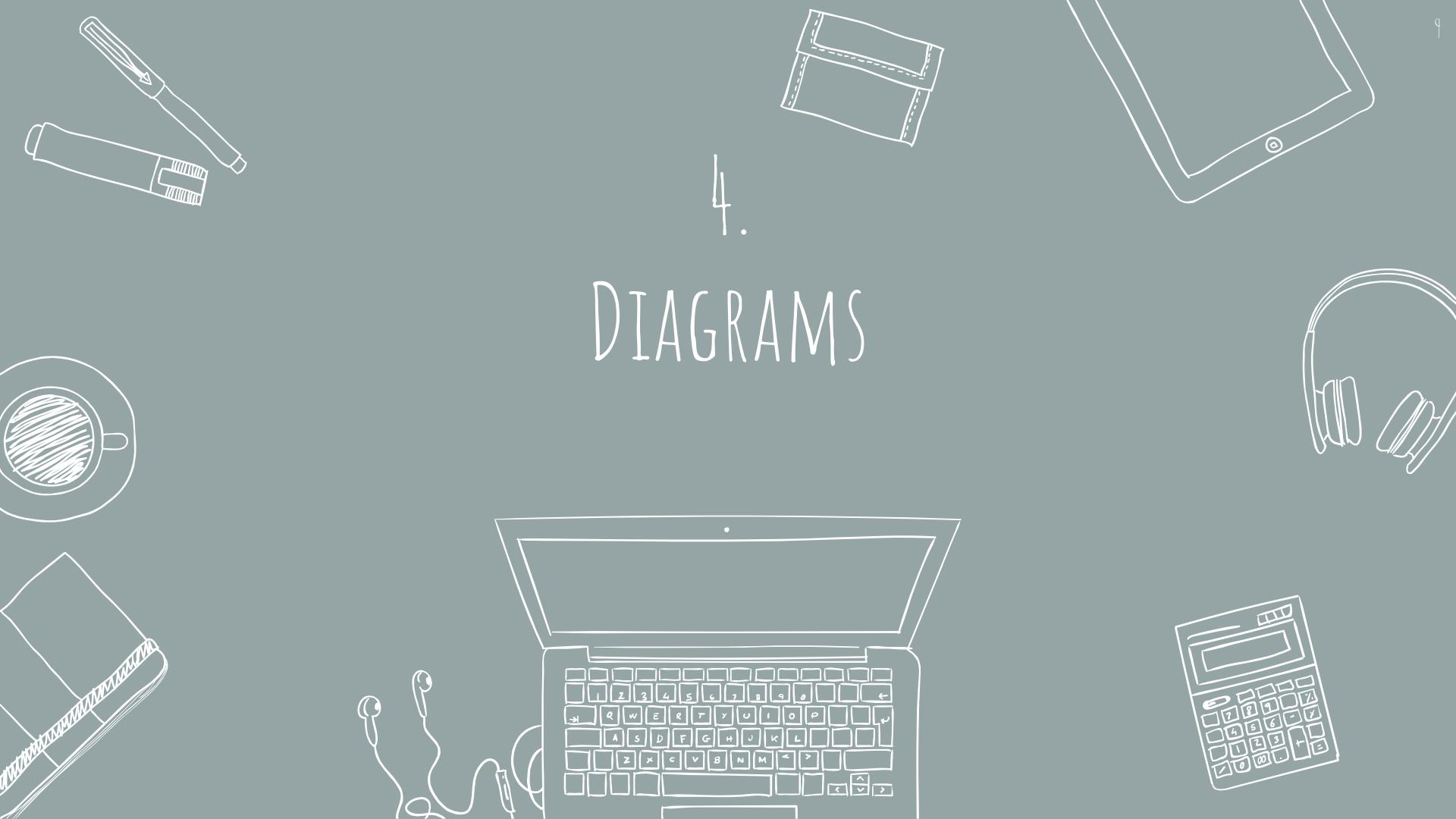
به طور متوسط در چه موقعی هنگام مراجعه به اتاق اساتید، استاد موردنظر در دسترس نبوده است؟

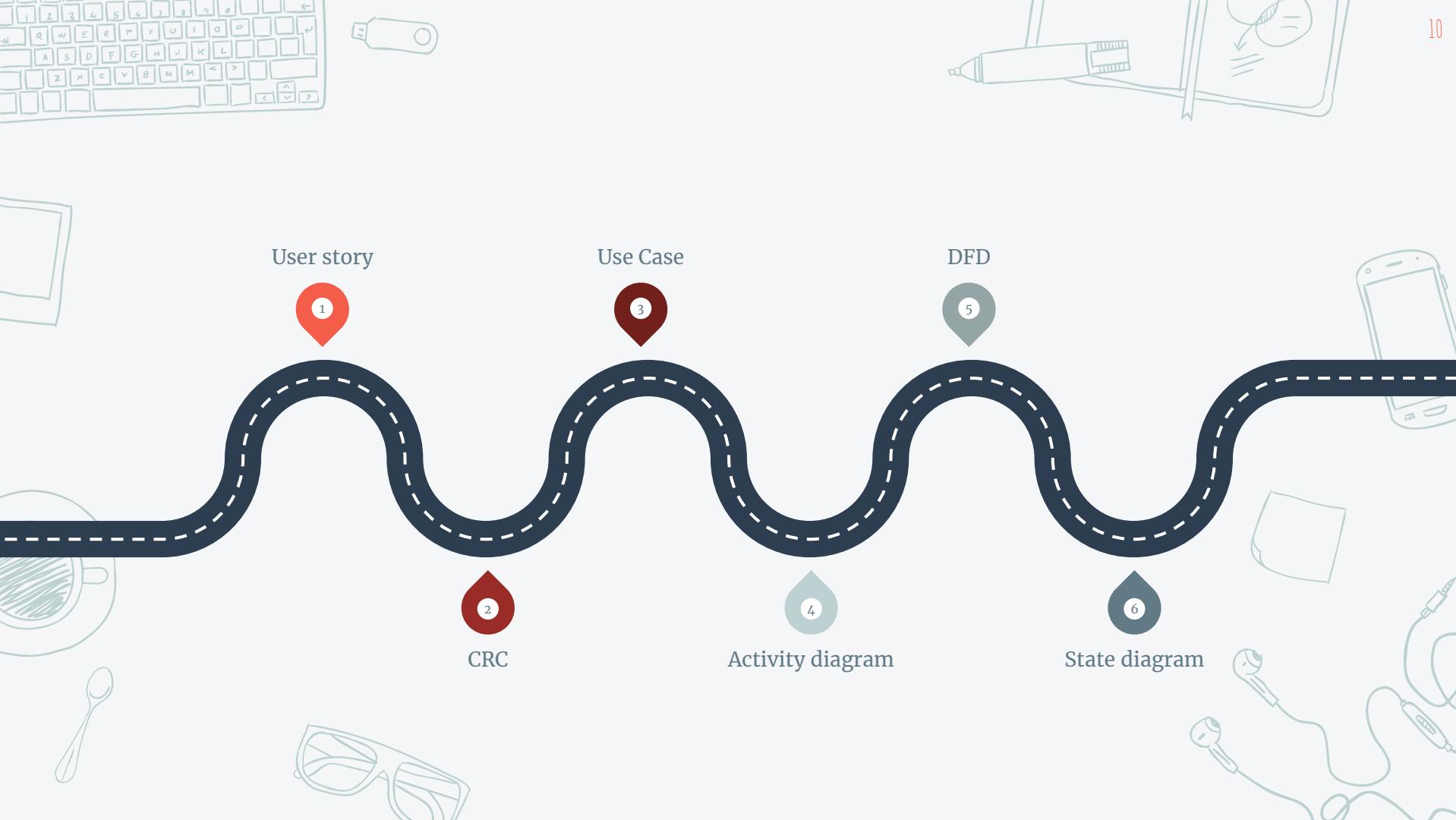
۲ پاسخ - (%)



4.

DIAGRAMS





USER STORY

ایده پیاده سازی یک اپلیکیشن برای **رزرو کردن تایم** با **اساتید** میتواند تایم زیادی از ما **دانشجویان دانشکده** سیو کند. به عنوان مثال اکثر اساتید هنگام مراجعه یا در دفتر خود حضور ندارند و یا وقت کافی ندارند و مشغول کار های تحقیقاتی خود می باشند. اینکه دو طرف بتوانند یک **جلسه** را **قانونمند** در یک محیط سنت کنند دیگر جای بحثی برای برگزاری جلسه باقی نخواهند ماند. بسیاری از بچه ها از زمان های اساتید آگاه نیستند. اگر بدانند که چه زمانی برای آن ها مناسب هست و آزاد هستند قطعاً **برنامه ریزی** مناسب تری برای باقی کارها خواهند داشت. علاوه بر آن، حتی میتواند به **زمانبندی** دانشجو نیز کمک کند. به عنوان مثال اگر من نوعی بخواهم با چند استاد میتینگ داشته باشم در صورت امکان سعی میکنم همه ی جلساتم در یک **روز** انجام بشود. و این امر با آگاهی از **برنامه هفتگی اساتید** و قدرت اختیار من در انتخاب، میسر خواهد بود. وجود اپلیکیشنی که قابلیتی را فراهم کند که یک کاربر متوجه شود چه جلساتی دارد و بتواند آن ها را بعد از رزرو **مشاهده** کند میتواند به جلوگیری از فراموش کردن آن ها کمک کند. یه نکته مهم دیگر هم به نظرم ارائه یک ساز و کار برای **پاییندی دولطف به شرکت** در جلسات است. در نهایت انتظارات من به عنوان یک یوزر از این اپلیکیشن این است که بتوانم جلسه خود را با استاد مدنظرم سنت کنم نه با زمانبندی هایی که قبلاً پشت در دفتر ذکر شده و برای چند ترم قبل است بلکه با **زمانبندی های تازه و آپدیت**.

USER STORY

در این برنامه ما انتظار داریم که بتوانیم به سهولت بهتری **زمانی** که خالی هستیم را اخذ کنیم. برای اینکه بتوانیم **جلساتی** که باید برویم را بررسی و در دسترس داشته باشیم، بهتر است **لیستی از جلسات داشته باشیم** و به این منظور باید هر کس یک **پروفایلی** برای ورود و ثبت نام داشته باشد.

بدین منظور به طرح زیر عمل می کنیم.

ابتدا وارد می شویم، یعنی اطلاعات مورد نیاز خود را برای ورود می زنیم. سپس وارد شده و لیست جلساتی را می بینیم و سپس بر حسب نیاز استادی را انتخاب می کنیم و **لیست زمان های او را می بینیم** و اشتراک بین زمان های خود را و او را در نظر می گیریم.

زمان مورد نظر خود را انتخاب می کنیم و جلسه اخذ می شود. می توانیم در صفحه دیگر ببینیم، که چه جلساتی در آینده خواهیم داشت.

MH STORY

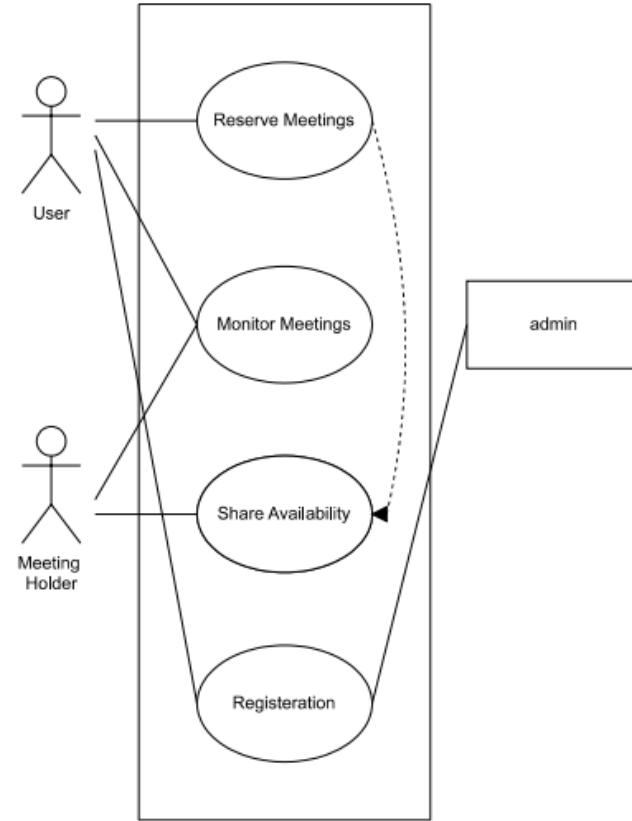
به نظرم در مرحله اول این نرم افزار باید user friendly باشد تا **افراد** مایل به استفاده از این ابزار باشند. به لحاظ کارکردی هم مایلم **جلساتی که درآینده دارم را ببینم**. این اهمیت دارد که افراد خود را ملزم به حضور در **جلسه های رزرو شده** ای خود بدانند. به علاوه خوب هست که سیستمی برای **ارزیابی افراد** در این خصوص وجود داشته باشد.

به طور کلی انتظار می رود **برنامه ریزی** را برای من علی الخصوص در **تخصیص زمان** جلسه در زمان های غیر از مطالعه و تحقیق راحت تر کند.

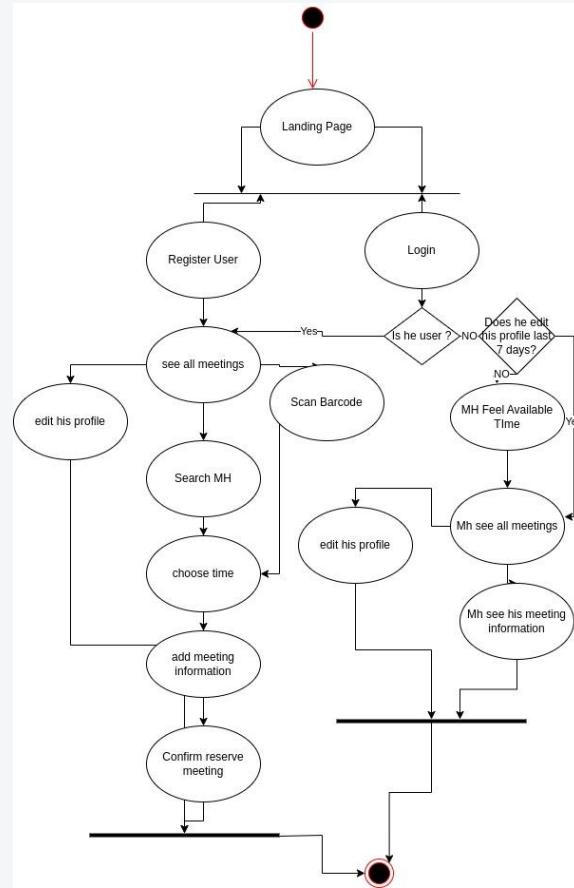
CRC

User	
<ul style="list-style-type: none">• contains user's attributes.• Reseve meetings with Meeting-Holders.• Show user's meetings.	<ul style="list-style-type: none">• MH• Meeting
MH	
<ul style="list-style-type: none">• Defines her\his free times.• Holdes meetings for users.	<ul style="list-style-type: none">• User• Meeting• Time
Meeting	
<ul style="list-style-type: none">• Keep meeting's attributes.	<ul style="list-style-type: none">• User• MH• Time
Time	
<ul style="list-style-type: none">• keep free time of Meeting-Holders.• Meetings will be held at these times.	<ul style="list-style-type: none">• Meeting• MH

USE CASE

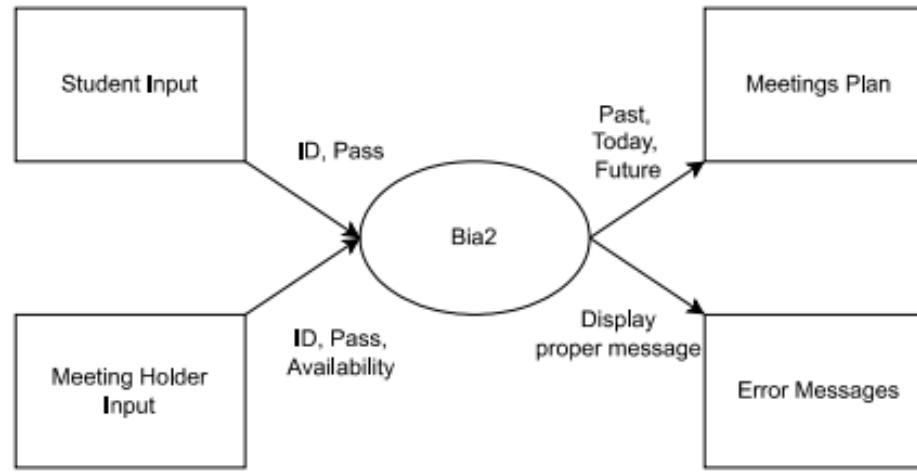


ACTIVITY DIAGRAM



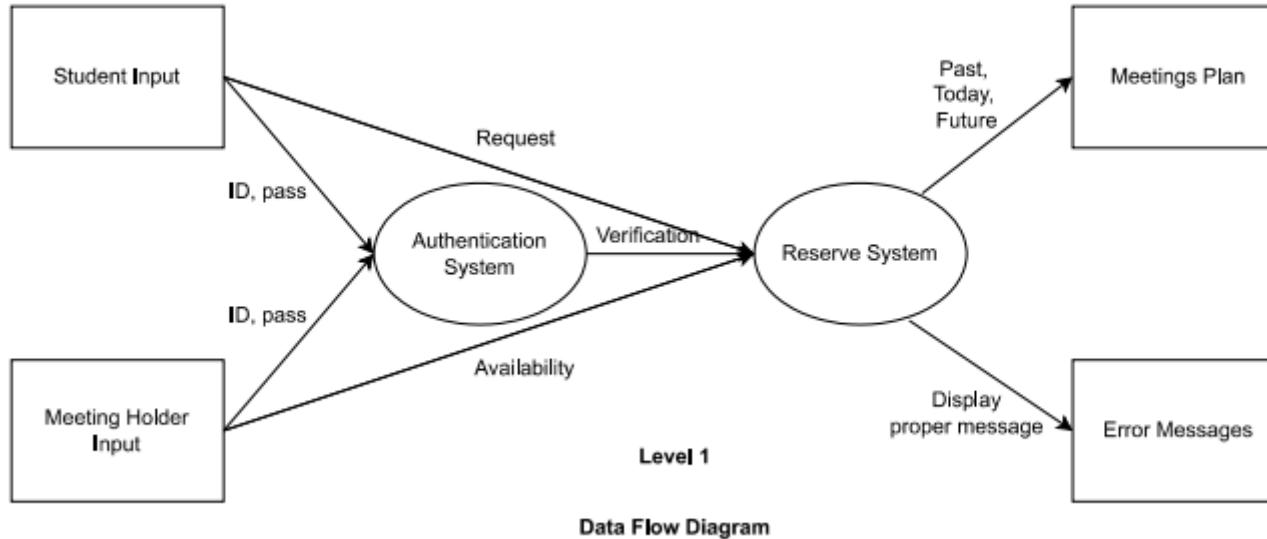
LEVEL 0

DFD

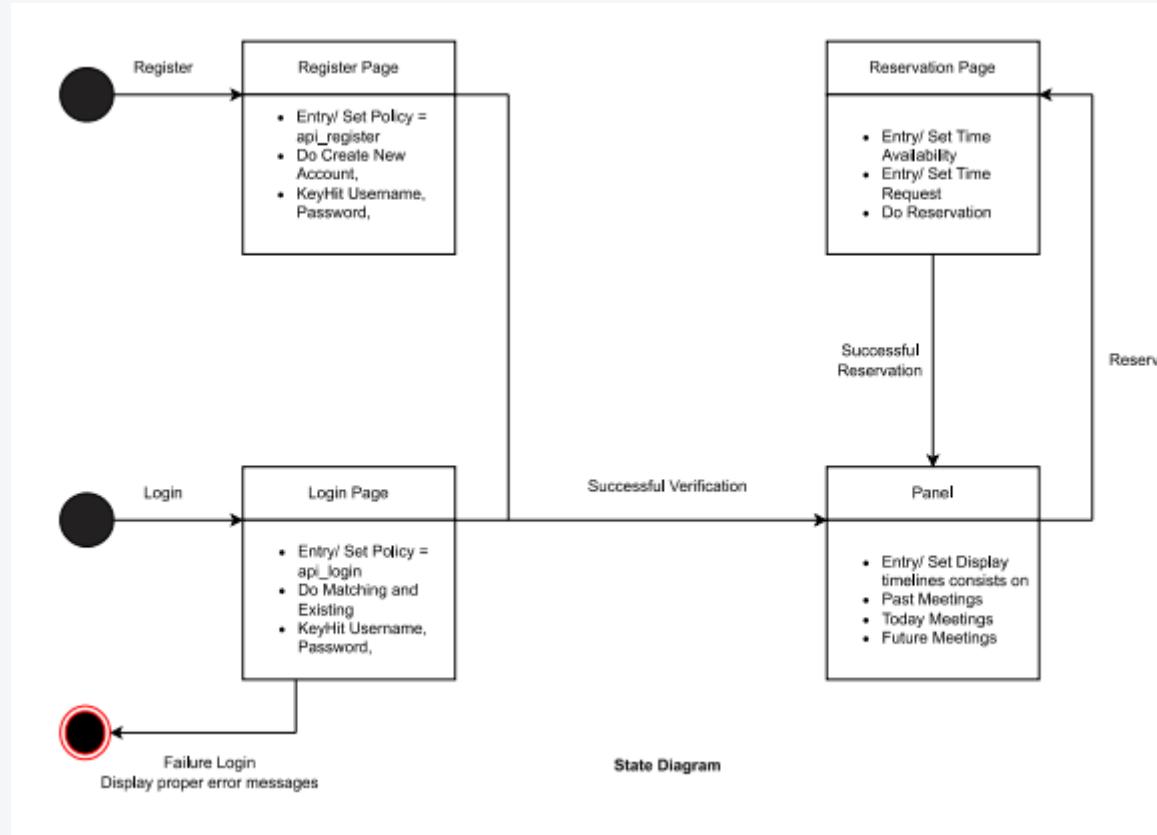


DFD

LEVEL 1



STATE DIAGRAM



Backend Frontend

Planning



6.

PROGRESS

- Weekly Sessions on Saturdays or mondays
- Daily Sessions (30 minute)
- Backend Team Sessions
- Frontend Team Sessions

FRONTEND DEVELOPMENT

Using React framework and Javascript

A/B TEST

The image displays two side-by-side screenshots of a mobile application's login interface. Both screens show a background image of a university campus with green buildings and trees. At the top left is a purple hand icon holding a red heart, and at the top right is a green hand icon clapping.

Left Screen (Control):

- Header:** ایمیل (Email)
- Text Input:** ایمیل خود را وارد کنید (Enter your email address). Below it: ایمیل دانشگاهی خود را وارد کنید (Enter your university email address).
- Password Input:** رمز عبور (Password). Below it: رمز عبور خود را وارد کنید (Enter your password).
- Remember Me:** مرا به خاطر بسیار (Remember me).
- Buttons:** **ورود** (Log In) and **ثبت نام** (Sign Up).

Right Screen (Variant):

- Header:** ایمیل (Email)
- Text Input:** ایمیل خود را وارد کنید (Enter your email address). Below it: ایمیل دانشگاهی خود را وارد کنید (Enter your university email address).
- Password Input:** رمز عبور (Password). Below it: رمز عبور خود را وارد کنید (Enter your password).
- Remember Me:** مرا به خاطر بسیار (Remember me).
- Buttons:** **ورود** (Log In) and **ثبت نام** (Sign Up).

IMPRESSIONS



COMMENTS:



Nesa Pabakht Zadeh • 2nd
Digital Product Designer (Ux Specialist)

2d (edited) ***

جند مورد در جهت بهبود بهتون بیشهاد صدم جون متشکل دسترسی بذیری داره
این طرح.

ایمیل راهنمایی که یاین نوشتید رو بوجود باکس بدارید. فضای استیت ارور رو
گرفقین با این دیراين.

رمز عبور نیازمند گذاشتن ایکن چشمکه که کاربر اگر خواست رمز رو ببینه.

وضعیت دکمه ها ناید بکسان باشه همواره یک دکمه مهمتره و باید اولویت
مشخص بشنه.

تیک منو به حاطر بسیار دوره.

فضای تنفس طرح کمه.

عکس پس زمینه حلیں شلوغه و باید روشن به کاور ساد که تصریک رو نگیره.

[See translation](#)

Like • 49 | Reply • 3 Replies

Discovery

22,491

Impressions

Engagements

Reactions

229 →

Comments

17 →

Reposts

0 →

Top demographics of people reached

Job titles 

Software Engineer · 407



Research Assistant · 404



Frontend Developer · 368

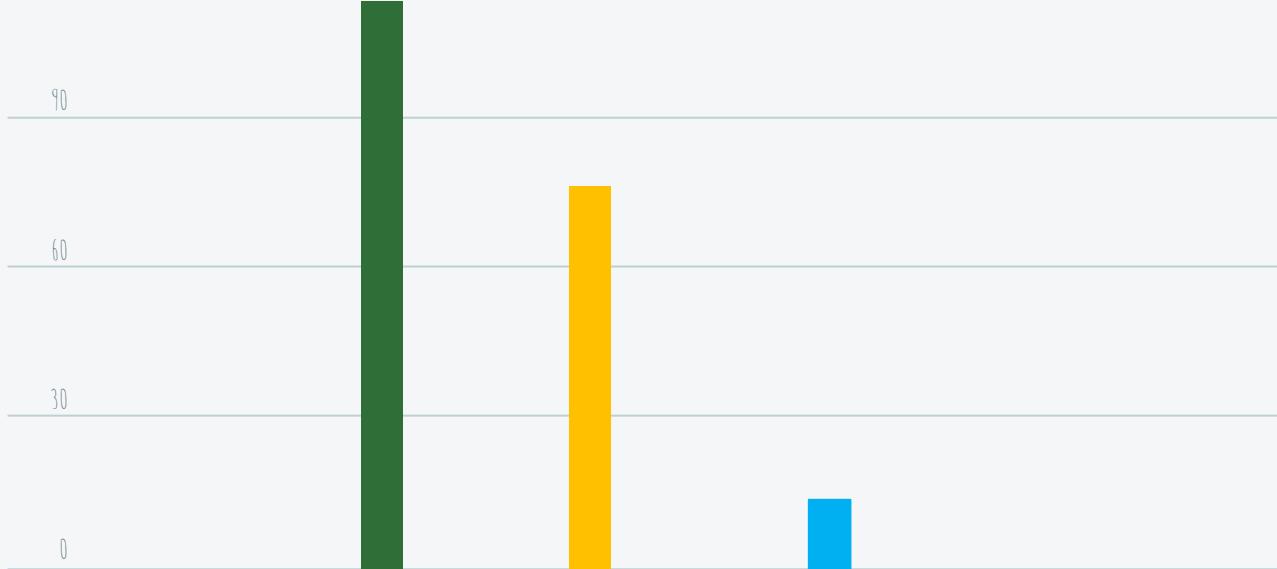


Teaching Assistant · 271

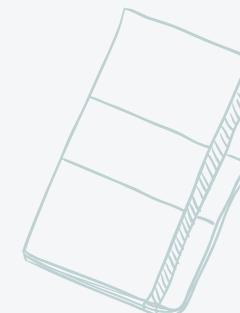


Chief Executive Officer · 240



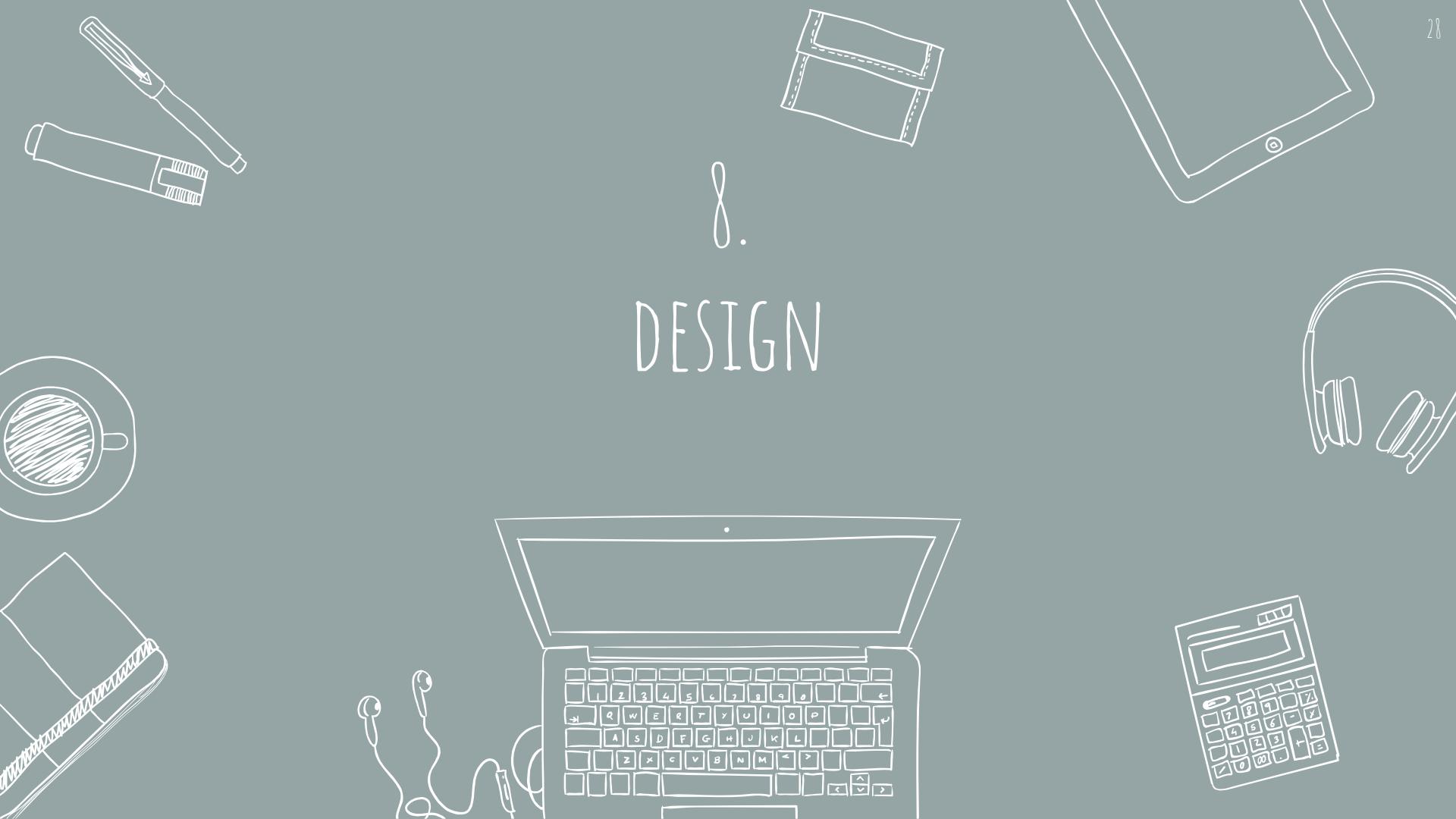


RESULT

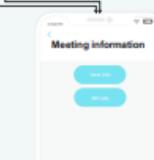
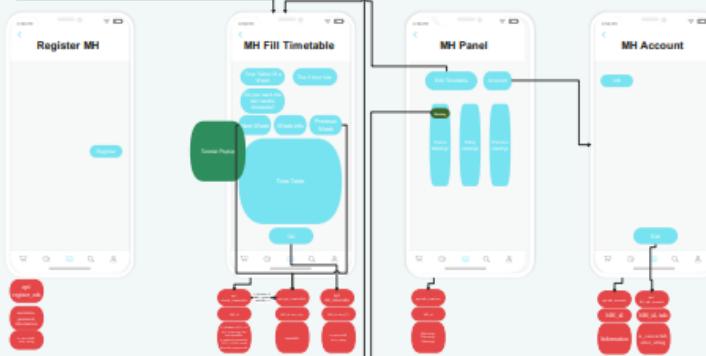
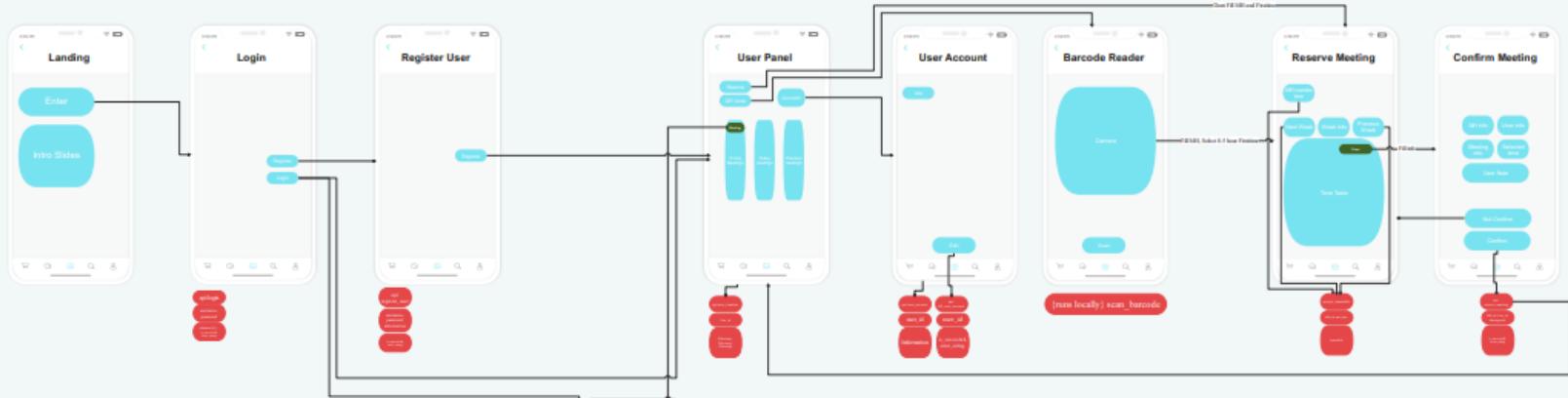


8.

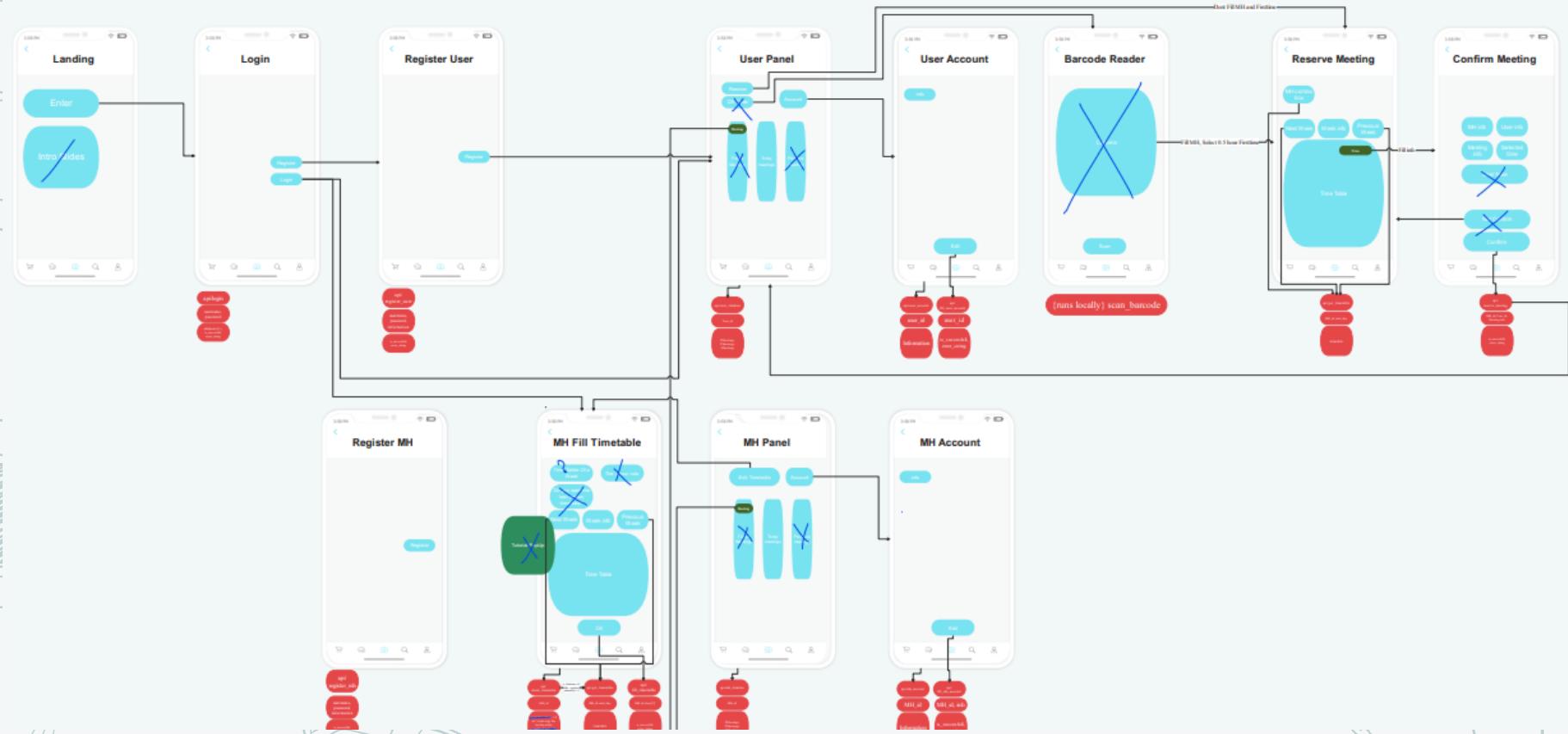
DESIGN

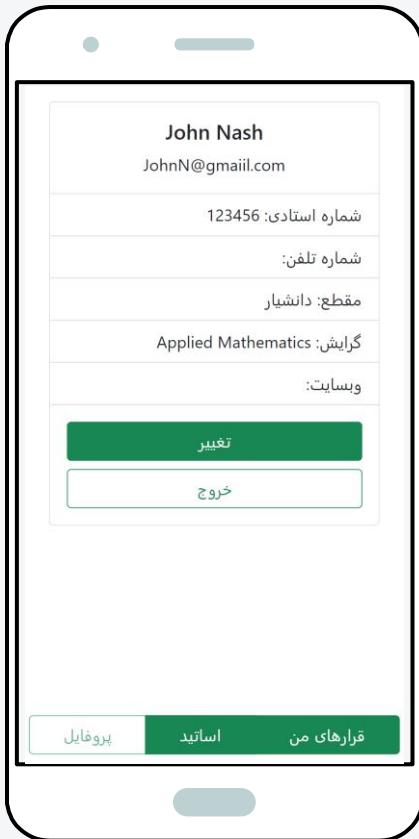


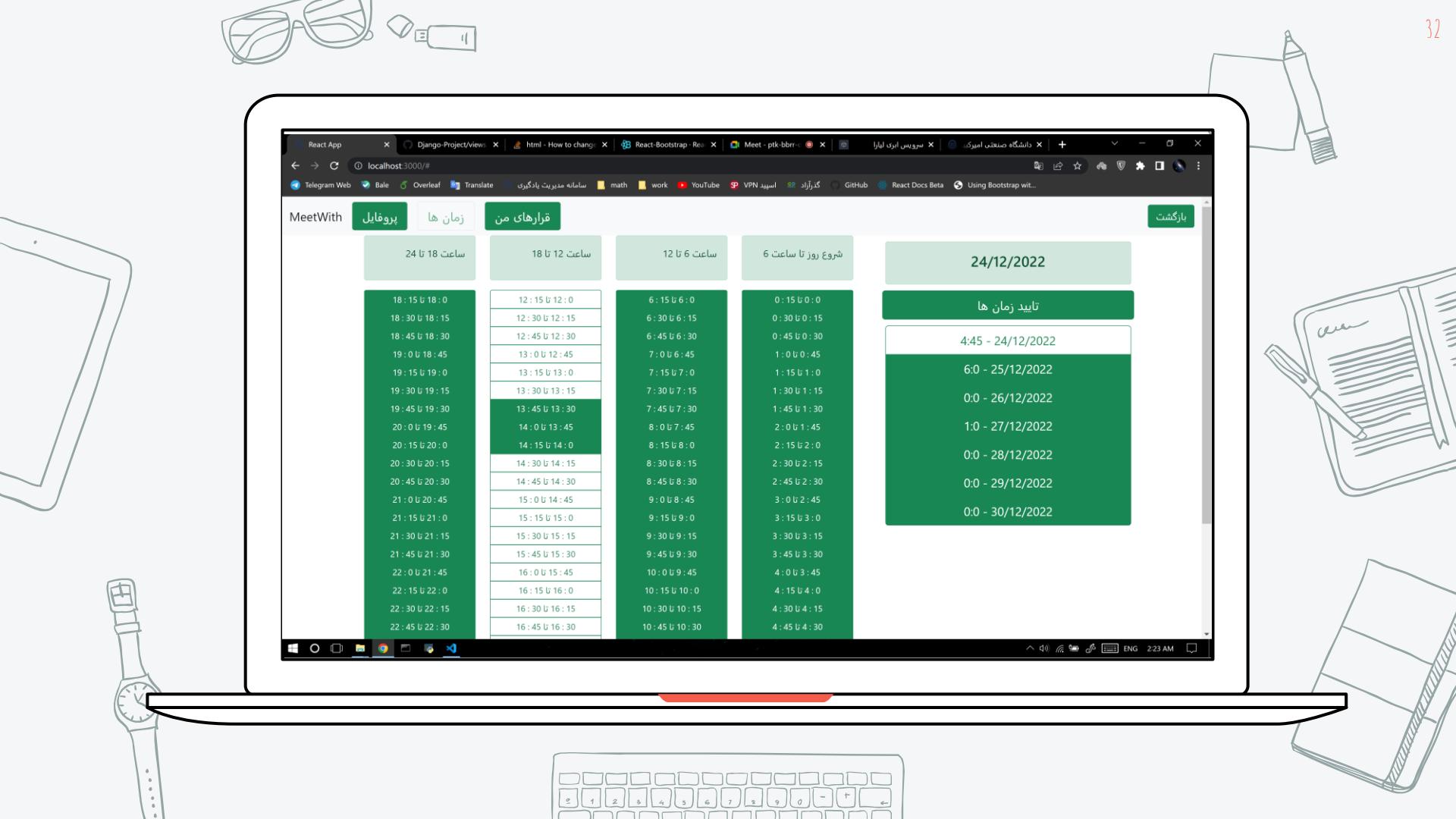
FINAL DESIGN



MVP

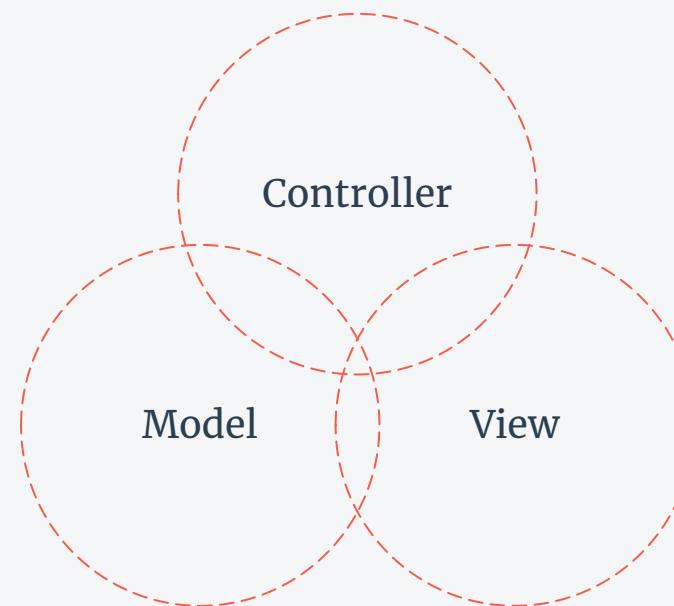






DESIGN PATTERN

DESIGN PATTERN



MVC

10. ORM



11.

BACKEND DEVELOPMENT

Using django framework and python



+1200

Lines of code

70

Commits in 4 branches.

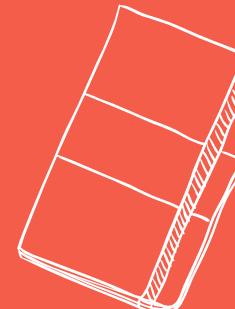
43

Days from first to last commit.

4

People contributed in backend

(one of our team members was working on frontend)



General overview of github commits

(HEAD -> master)	commit	access levels added - get/set accounts api's added - some bugs fixed.	bia2django updated!	Author: AbolfazlAAnsari <abolfazlansari.aut@gmail.com>	mh_account, and user_account adapted	Author: AbolfazlAAnsari <abolfazlansari.aut@gmail.com>
Author: bateni1380 <bateni1380@gmail.com>	Author: bateni1380 <bateni1380@gmail.com>	Date: Mon Dec 19 13:16:21 2022 +0330	commit	Author: bateni1380 <bateni1380@gmail.com>	Date: Thu Dec 15 20:26:55 2022 +0330	Author: AbolfazlAAnsari <abolfazlansari.aut@gmail.com>
Date: Sat Dec 24 01:40:47 2022 +0330				Author: bateni1380 <bateni1380@gmail.com>	Date: Wed Dec 14 15:20:18 2022 +0330	Test commit
bia2django updated!	access levels added - get/set accounts api's added - some bugs fixed.	access levels added - get/set accounts api's added - some bugs fixed.	register_mh_and_register_user_and_login_api's has finished.	database configuration with larva postgresql	some adaptation on api login	commit
commit origin/HEAD)						Author: NeginR <70764388+NeginR@users.noreply.github.com>
Author: bateni1380 <bateni1380@gmail.com>	Author: bateni1380 <bateni1380@gmail.com>	Date: Mon Dec 19 02:38:35 2022 +0330	commit	Author: nekoouu <2120061159+nekoouu@users.noreply.github.com>	Date: Tue Dec 13 01:17:48 2022 +0330	including codes Related to ORM
Date: Fri Dec 23 14:03:01 2022 +0330				Author: nekoouu <2120061159+nekoouu@users.noreply.github.com>	Date: Sat Dec 10 11:46:57 2022 +0330	
process time added to each api	access levels added - get/set accounts api's added - some bugs fixed.	access levels added - get/set accounts api's added - some bugs fixed.	api_mh_timeline updated.	Add files via upload	list of mh api	commit
						Author: nekoouu <2120061159+nekoouu@users.noreply.github.com>
commit	Author: bateni1380 <bateni1380@gmail.com>	Date: Sun Dec 18 22:18:27 2022 +0330	commit	Author: bateni1380 <bateni1380@gmail.com>	Date: Mon Dec 12 13:37:17 2022 +0330	Delete ORM
Author: bateni1380 <bateni1380@gmail.com>	Author: bateni1380 <bateni1380@gmail.com>	Date: Fri Dec 23 13:56:21 2022 +0330	commit	Author: nekoouu <2120061159+nekoouu@users.noreply.github.com>	Date: Wed Dec 14 13:37:12 2022 +0330	its the body of our django project!
bugs fixed!				Author: nekoouu <2120061159+nekoouu@users.noreply.github.com>	Date: Fri Dec 9 20:55:34 2022 +0330	
commit	Author: bateni1380 <bateni1380@gmail.com>	Date: Sun Dec 18 16:26:48 2022 +0330	api_mh_timeline updated.	Add files via upload	All of API's functions are located in bia2django/api/views	Related to ORM
Author: bateni1380 <bateni1380@gmail.com>	Author: bateni1380 <bateni1380@gmail.com>	Date: Fri Dec 23 13:54:37 2022 +0330	commit	Author: bateni1380 <bateni1380@gmail.com>	Date: Mon Dec 12 13:37:17 2022 +0330	
bugs fixed!				Author: nekoouu <2120061159+nekoouu@users.noreply.github.com>	Date: Wed Dec 14 13:37:12 2022 +0330	
commit	Author: bateni1380 <bateni1380@gmail.com>	Date: Sun Dec 18 16:26:48 2022 +0330	api_mh_timeline has been added!	model and ORM functions	TEST2	commit
Author: bateni1380 <bateni1380@gmail.com>	Author: bateni1380 <bateni1380@gmail.com>	Date: Fri Dec 23 13:54:37 2022 +0330	commit	Author: bateni1380 <bateni1380@gmail.com>	Date: Mon Dec 12 13:37:17 2022 +0330	Author: nekoouu <2120061159+nekoouu@users.noreply.github.com>
bugs fixed!				Author: nekoouu <2120061159+nekoouu@users.noreply.github.com>	Date: Wed Dec 14 13:37:12 2022 +0330	commit
commit	Author: bateni1380 <bateni1380@gmail.com>	Date: Sun Dec 18 16:46:55 2022 +0330	api_mh_timeline has been added!	Delete models.py	TEST2	commit
Author: bateni1380 <bateni1380@gmail.com>	Author: bateni1380 <bateni1380@gmail.com>	Date: Fri Dec 23 13:50:37 2022 +0330	commit	Author: AbolfazlAAnsari <abolfazlansari.aut@gmail.com>	Date: Mon Dec 12 13:37:17 2022 +0330	Author: nekoouu <2120061159+nekoouu@users.noreply.github.com>
bugs fixed!				Author: bateni1380 <bateni1380@gmail.com>	Date: Wed Dec 14 13:37:12 2022 +0330	commit
commit	Author: bateni1380 <bateni1380@gmail.com>	Date: Sun Dec 18 15:46:55 2022 +0330	comments of views.py has been added!	Delete api/ORM directory	commit	commit
Author: bateni1380 <bateni1380@gmail.com>	Author: bateni1380 <bateni1380@gmail.com>	Date: Fri Dec 23 13:50:37 2022 +0330	commit	Author: AbolfazlAAnsari <abolfazlansari.aut@gmail.com>	Date: Mon Dec 12 13:37:17 2022 +0330	Author: nekoouu <2120061159+nekoouu@users.noreply.github.com>
bugs fixed!				Author: bateni1380 <bateni1380@gmail.com>	Date: Wed Dec 14 13:37:12 2022 +0330	commit
commit	Author: bateni1380 <bateni1380@gmail.com>	Date: Sun Dec 18 15:46:55 2022 +0330	comments of views.py has been added!	Payam	TEST	commit
Author: bateni1380 <bateni1380@gmail.com>	Author: bateni1380 <bateni1380@gmail.com>	Date: Fri Dec 23 13:44:38 2022 +0330	comments of views.py has been added!	in login api, two checkers func of database side were added	Author: AbolfazlAAnsari <abolfazlansari.aut@gmail.com>	commit
bugs fixed!				Author: bateni1380 <bateni1380@gmail.com>	Date: Wed Dec 14 03:44:09 2022 +0330	Author: nekoouu <2120061159+nekoouu@users.noreply.github.com>
commit	Author: bateni1380 <bateni1380@gmail.com>	Date: Sun Dec 18 14:47:45 2022 +0330	comments of views.py has been added!	commit	Author: AbolfazlAAnsari <abolfazlansari.aut@gmail.com>	commit
Author: bateni1380 <bateni1380@gmail.com>	Author: bateni1380 <bateni1380@gmail.com>	Date: Fri Dec 23 13:44:38 2022 +0330	register_mh, register_user, login, mh_fill_timetable, reserve_meeting, get_timetable api's are completed!	commit	Author: AbolfazlAAnsari <abolfazlansari.aut@gmail.com>	commit
bugs fixed!				Author: bateni1380 <bateni1380@gmail.com>	Date: Wed Dec 14 03:44:09 2022 +0330	Author: nekoouu <2120061159+nekoouu@users.noreply.github.com>
commit	Author: bateni1380 <bateni1380@gmail.com>	Date: Sun Dec 18 14:47:45 2022 +0330	register_mh, register_user, login, mh_fill_timetable, reserve_meeting, get_timetable api's are completed!	bia2django updated!	user_timeline and mh_timeline have done	commit
Author: bateni1380 <bateni1380@gmail.com>	Author: bateni1380 <bateni1380@gmail.com>	Date: Fri Dec 23 13:25:36 2022 +0330	commit	Author: nekoouu <2120061159+nekoouu@users.noreply.github.com>	Author: AbolfazlAAnsari <abolfazlansari.aut@gmail.com>	delete comment note
two policies added in views.				Author: nekoouu <2120061159+nekoouu@users.noreply.github.com>	Date: Tue Dec 13 21:32:40 2022 +0330	commit
commit	Author: bateni1380 <bateni1380@gmail.com>	Date: Sun Dec 18 01:25:56 2022 +0330	updated.	Author: nekoouu <2120061159+nekoouu@users.noreply.github.com>	Date: Wed Dec 14 17:18:11 2022 +0330	commit
Author: bateni1380 <bateni1380@gmail.com>	Author: bateni1380 <bateni1380@gmail.com>	Date: Fri Dec 23 13:25:36 2022 +0330	bia2django updated!	fill_user_account and fill_mh_account api	Author: NeginR <70764388+NeginR@users.noreply.github.com>	upload all Mr.Arabansari's apis.
commit				Author: nekoouu <2120061159+nekoouu@users.noreply.github.com>	Date: Sat Dec 10 20:51:09 2022 +0330	commit
commit	Author: bateni1380 <bateni1380@gmail.com>			Author: nekoouu <2120061159+nekoouu@users.noreply.github.com>	Date: Fri Dec 9 11:21:52 2022 +0330	commit
commit				Author: nekoouu <2120061159+nekoouu@users.noreply.github.com>	Date: Sat Dec 10 20:52:32 2022 +0330	commit
commit				Author: nekoouu <2120061159+nekoouu@users.noreply.github.com>	Date: Mon Dec 11 20:30:14 2022 +0330	commit
commit				Author: nekoouu <2120061159+nekoouu@users.noreply.github.com>	Date: Fri Dec 9 09:30:14 2022 +0330	commit
commit				Author: nekoouu <2120061159+nekoouu@users.noreply.github.com>	Date: Sat Dec 10 20:49:55 2022 +0330	commit
commit				Author: nekoouu <2120061159+nekoouu@users.noreply.github.com>	Date: Sat Dec 10 20:51:09 2022 +0330	commit
commit				Author: nekoouu <2120061159+nekoouu@users.noreply.github.com>	Date: Fri Dec 9 09:32:15 2022 +0330	commit
commit				Author: nekoouu <2120061159+nekoouu@users.noreply.github.com>	Date: Fri Dec 9 09:32:15 2022 +0330	add information for mb

Project files:

```

  ↘ bia2django
    ↘ api
      > __pycache__
      > migrations
      ⚭ __init__.py
      ⚭ admin.py
      ⚭ apps.py
      ⚭ controller.py
      ⚭ main_functions.py
      ⚭ models.py
      ⚭ tests.py
      ⚭ urls.py
      ⚭ views.py
    ↘ bia2django
      > __pycache__
      ⚭ __init__.py
      ⚭ asgi.py
      ⚭ settings.py
      ⚭ urls.py
      ⚭ wsgi.py
    ≡ db.sqlite3
    ⚭ manage.py
    ≡ requirements.txt

```

bia2django/bia2django/urls.py

```

19 ↘ urlpatterns = [
20     path('admin/', admin.site.urls),
21     path('api/', include('api.urls'))
22 ]
23

```

bia2django/api/urls.py

```

4   # URLConf
5 ↘ urlpatterns = {
6     path('register_user/', views.api_register_user),
7     path('register_mh/', views.api_register_mh),
8     path('login/', views.api_login),
9     path('mh_fill_timetable/', views.api_mh_fill_timetable),
10    path('get_list_of_mh/', views.api_get_list_of_mh),
11    path('reserve_meeting/', views.api_reserve_meeting),
12    path('get_timetable/', views.api_get_timetable),
13    path('get_mh_timeline/', views.api_get_mh_timeline),
14    path('get_user_timeline/', views.api_get_user_timeline),
15    path('get_mh_account/', views.api_get_mh_account),
16    path('get_user_account/', views.api_get_user_account),
17    path('fill_mh_account/', views.api_fill_mh_account),
18    path('fill_user_account/', views.api_fill_user_account),
19 }

```



Database structure (models.py)

```
class Time(models.Model):
    date = models.DateField()
    start_time = models.TimeField()
    end_time = models.TimeField()
```

```
class User(models.Model):
    first_name = models.CharField(max_length= 100)
    last_name = models.CharField(max_length= 100)
    user_email = models.EmailField(max_length=100)
    user_password = models.CharField(max_length=100)
    student_number = models.CharField(max_length=50)
    mobile_number = models.CharField(max_length= 50)
    degree = models.CharField(max_length= 100)
    field = models.CharField(max_length= 100)
    university = models.CharField(max_length= 100)
    adviserID = models.ForeignKey(MH, on_delete=models.CASCADE)
```

```
class Meeting(models.Model):
    subject = models.CharField(max_length= 100,default=None)
    rate = models.IntegerField(default=None)
    description = models.TextField(default=None)
    was_holded = models.BooleanField(default=None)
    mhID = models.ForeignKey(MH, on_delete=models.CASCADE)
    userID = models.ForeignKey(User, on_delete=models.CASCADE)
    timeID = models.ForeignKey(Time, on_delete=models.CASCADE)
```

```
class MH(models.Model):
    first_name = models.CharField(max_length= 100)
    last_name = models.CharField(max_length= 100)
    mh_email = models.EmailField(max_length=100)
    mh_password = models.CharField(max_length=100)
    teacher_number = models.CharField(max_length=50)
    degree = models.CharField(max_length= 100)
    field = models.CharField(max_length= 100)
    link_to_webpage = models.CharField(max_length= 1000)
```

```
class MH_Time(models.Model) :
    mhID = models.ForeignKey(MH, on_delete=models.CASCADE)
    timeID = models.ForeignKey(Time, on_delete=models.CASCADE)
```

Need of ORM functions in main_functions.py

```
1 # Used in: register_user
2 from api.controller import get_role, add_user
3 # Used in: register_mh
4 from api.controller import get_role, add_mh
5 # Used in: login
6 from api.controller import get_role, check_password, get_user_id
7 # Used in: mh_fill_timetable
8 from api.controller import remove_mh_times, append_time, append_mh_time
9 # Used in: get_list_of_mh
10 from api.controller import get_mh_list
11 # Used in: reserve_meeting
12 from api.controller import append_time, remove_time_from_mh_times, append_meeting
13 # Used in: get_timetaible
14 from api.controller import get_mh_times
15 # Used in: mh_timeline
16 from api.controller import mh_meetings
17 # Used in: user_timeline
18 from api.controller import user_meetings
19 # Used in: get_mh_account and get_user_account
20 from api.controller import get_account
21 # Used in: fill_mh_account and fill_user_account
22 from api.controller import set_account
```

Comments of one function in views.py

```
196 def api_login(request):
197     ...
198     This api logins a user and returns it's ID (if loggedin).
199     input: email, password
200     output: role, person_id, is_succesfull, error_string
201     api url: [host_address]/api/login/
202     access level: only the email owner - you need to provide the password that is assigned to the email
203     input example:
204     {
205         "email": "IsaacN@gmail.com",
206         "password": "87654321"
207     }
208     output example:
209     {
210         "role": "user",
211         "person_id": 1,
212         "is_succesfull": true,
213         "error_string": ""
214     }
215     ...
216     try:
217         start_time = time.time()
218         input = json.loads(request.body)
219         input = check_inputs(input, 'api_login')
```

Input validation for each api (in views.py)

```
18 def check_inputs(input, policy):
19     if policy == 'api_register_user':
20         if 'first_name' not in input: raise Exception("You must mention first name!")
21         if 'last_name' not in input: raise Exception("You must mention last name!")
22         if 'user_email' not in input: raise Exception("You must mention email!")
23         if 'user_password' not in input: raise Exception("You must mention password!")
24         elif len(input['user_password'])<8: raise Exception("Length of password must be at least 8!")
25         if 'student_number' not in input: raise Exception("You must mention student number!")
26         if 'mobile_number' not in input: raise Exception("You must mention mobile number!")
27         if 'degree' not in input: input['degree']=''
28         if 'field' not in input: input['field']=''
29         if 'university' not in input: input['university']=''
30         if 'adviserID' not in input: raise Exception("You must mention adviserID!")
31
32 >     if policy == 'api_register_mh':...
33
34 >     if policy == 'api_login':...
```



Calculating proper output for each api (in views.py)

```
try:  
    start_time = time.time()  
    input = json.loads(request.body)  
    input = check_inputs(input, 'api_register_user')  
    output = register_user(input)  
    output.update(succes)  
    output['process_time'] = time.time()-start_time  
except Exception as e:  
    output = {"is_succesfull": False, "error_string": str(e),}  
  
return JsonResponse(output)
```

THANKS!

Any questions?