

aapoon meet

Detailed Architecture

# Overview:

This document contains the details of the Architecture details like the Tech Stack and data models, some infrastructure details as well as some necessary details for Frontend. The data models are according to the requirement but they are not final as there is always possibility of addition of some fields, maybe because of some requirements and sometimes to increase the ease of access by introducing a little denormalization and data redundancy.

## Tech Stack:

MERN (**MongoDB, Express, ReactJs , Nodejs**) will be used for the project. Here are the reasons to choose:

- **Mongo:** NoSQL easy to scale, and comes with all the benefits of NoSQL
- **Express:** arguably the most reliable server to run Nodejs. It's fast, can handle load and thousands of concurrent requests with ease with the help of node's non-blocking structure.
- **ReactJs:** most popular frontend library today and it has very good develop community too. It's easy to create reusable components in react and with frameworks like material-ui, it works like a charm.
- **Nodejs:** Fast, reliable and easy to scale, relatively newer compared to technologies like .NET, PHP etc. Lesser prone to attacks since it runs behind a proxy instead of user directly requesting a script (like xyz.asp or xyz.php).

## Infrastructure Requirements:

The application does not impose any specific requirements to run, it has two parts frontend and backend and all the communication between frontend and backend is through APIs.

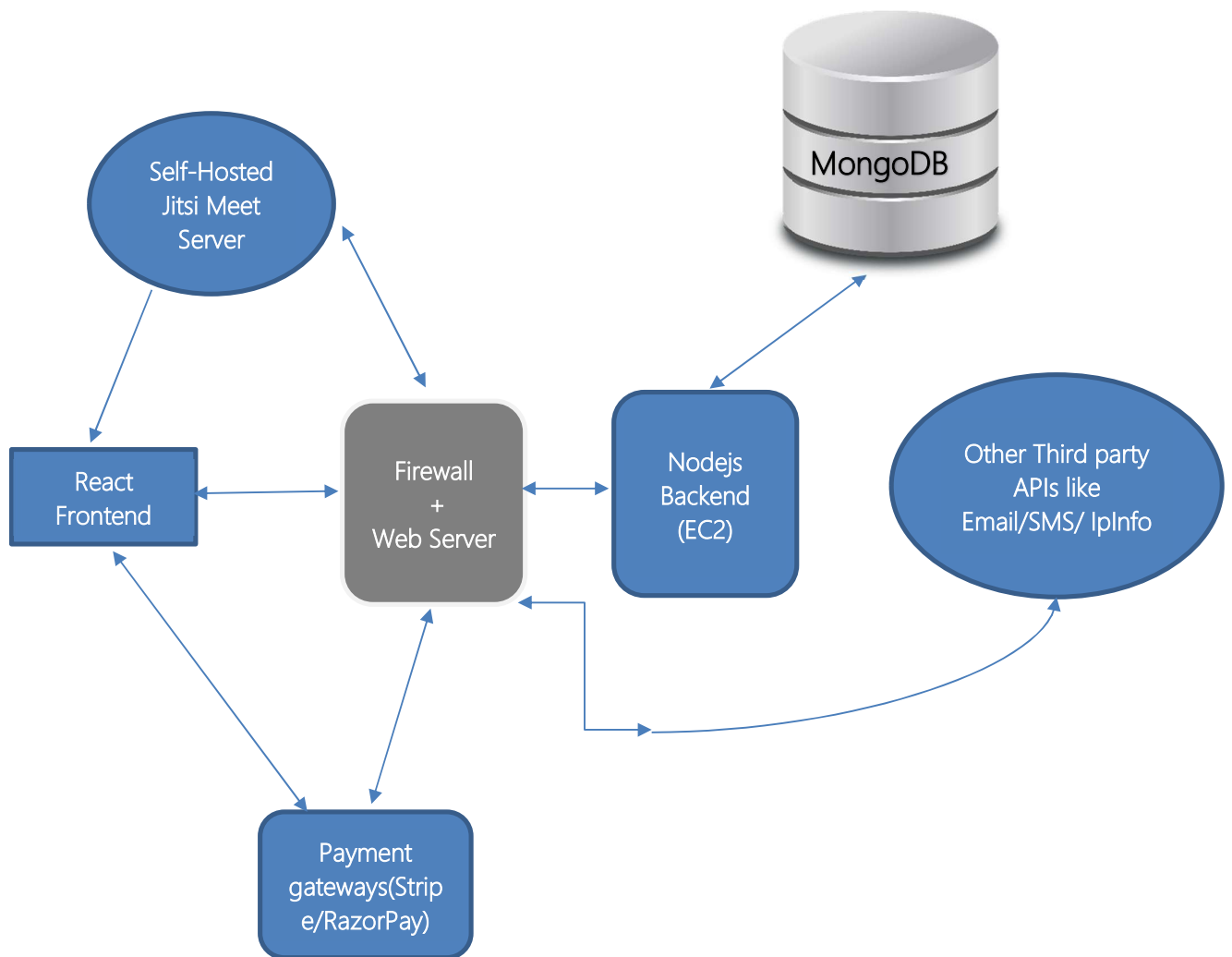
The frontend needs nothing but a **web server** to host it. It can be hosted on any web server, or on **AWS C2, S3, Azure Storage or similar file storages** as well.

The backend Needs a web server + a machine/container where nodejs files can run (means it has nodejs, npm etc. installed). It can be deployed on any **VPS, AWS services like EC2 (Reverse proxy from Nginx), lambda, Lightsail, Elastic Beanstack etc.** or similar services provided by other providers.

## User Authentication/Authorization:

- Token Based Authentication
- Token will be sent with every request in the Authorization header, for every request that requires Authentication.
- Token and passwords will not be saved, only the hashes will be saved and compared for security.

- JSON Web Token will be used



# **Database Model**

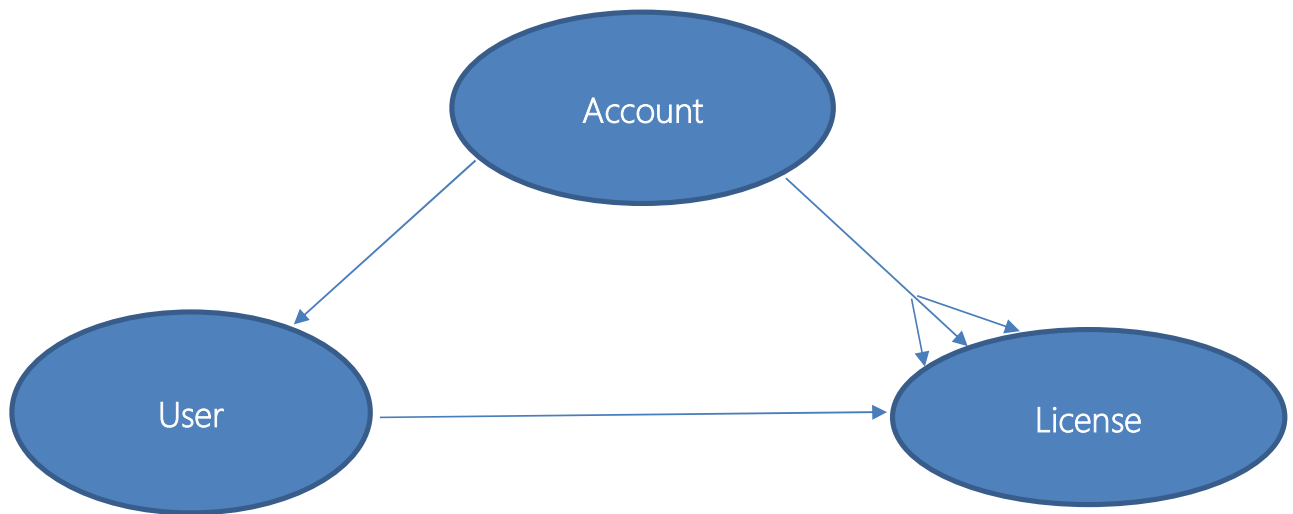
appoon@cluster0-shard-00-00-  
ug4xb.mongodb.net

## **Data Dictionary**

2021-11-17

# User & Account Management:

User is a separate entity & Account is separate, Account can be Professional or Business.




























## Table of contents

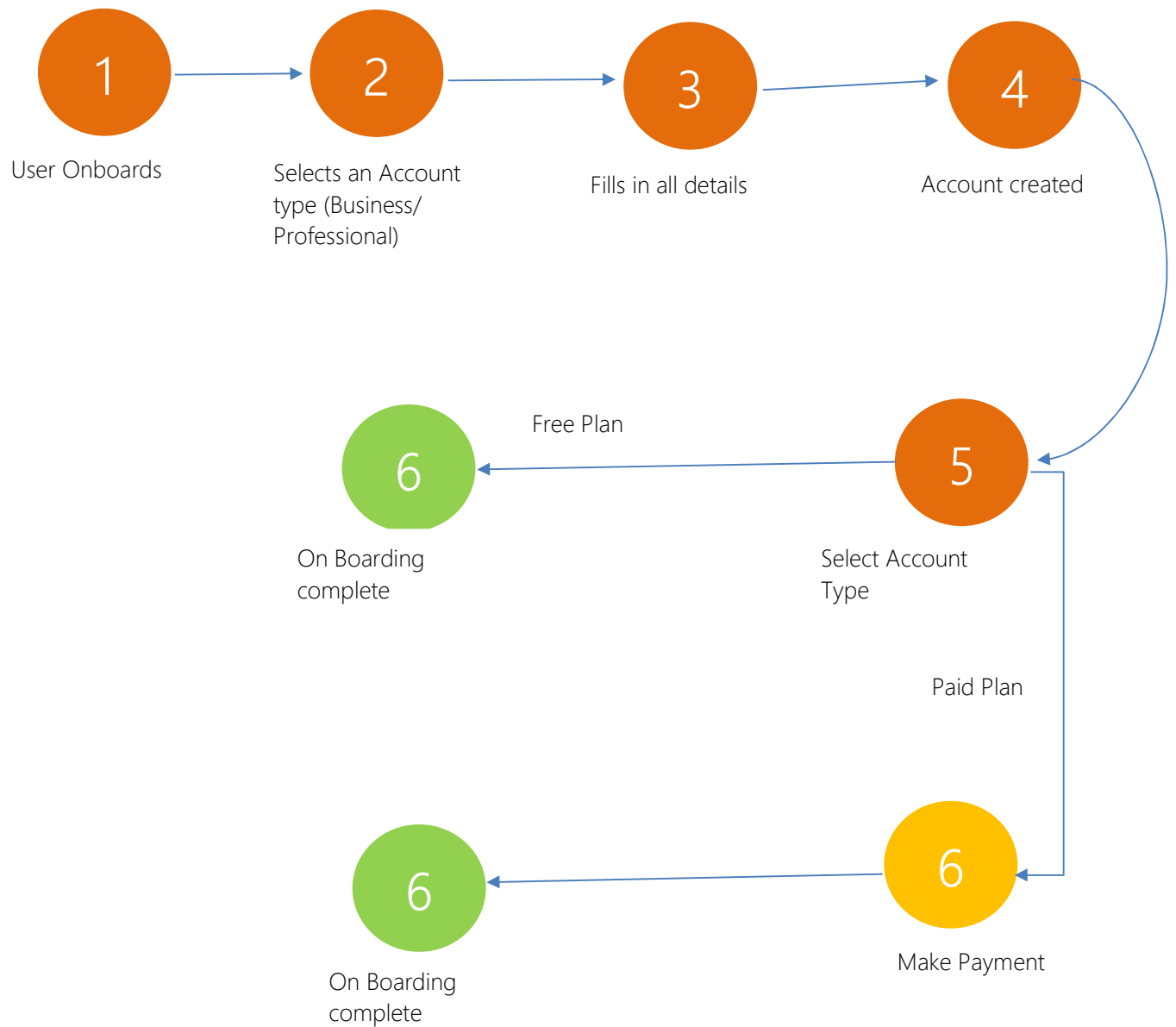
apsoon@cluster0-shard-00-00-ug4xb.mongodb.net .....	6
1. User registration .....	7
1.1. Tables .....	8
1.1.1. Collection: account (User account details) .....	8
1.1.2. Collection: license (User or business licenses) .....	9
1.1.3. Collection: users (Users schema) .....	10
2. Meeting scheduler .....	13
2.1. Tables .....	14
2.1.1. Collection: account (User account details) .....	14
2.1.2. Collection: meetings (All meetings data here) .....	15
2.1.3. Collection: users (Users schema) .....	16
3. Subscriptions .....	19
3.1. Tables .....	20
3.1.1. Collection: account (User account details) .....	20
3.1.2. Collection: invoice (All invoice generated) .....	21
3.1.3. Collection: license (User or business licenses) .....	22
3.1.4. Collection: payments (all inward and outward payment details) .....	23
3.1.5. Collection: users (Users schema) .....	24
4. user dashboard .....	27
5. Calendar .....	28
5.1. Tables .....	29
5.1.1. Collection: account (User account details) .....	29
5.1.2. Collection: meetings (All meetings data here) .....	30
5.1.3. Collection: users (Users schema) .....	31
6. Customer Admin Panel .....	34
7. User Invitation .....	35
7.1. Tables .....	36
7.1.1. Collection: account (User account details) .....	36
7.1.2. Collection: invitation (invitation link data with validity) .....	37
7.1.3. Collection: license (User or business licenses) .....	38
7.1.4. Collection: users (Users schema) .....	39
8. Other .....	42
8.1. Tables .....	42
8.1.1. Collection: staff (all staff data with roles here) .....	42

## Legend

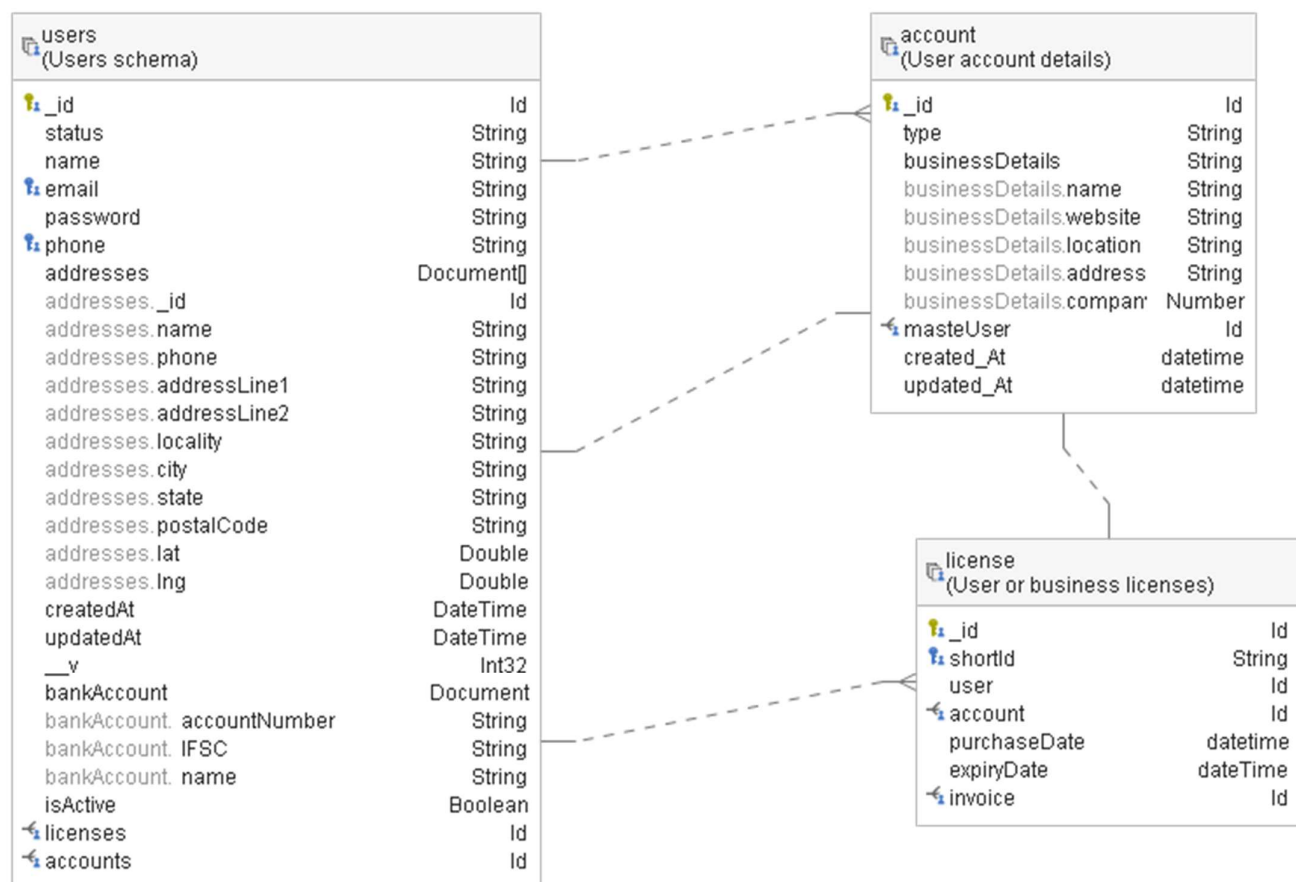
-  Primary key
-  Primary key disabled
-  User-defined primary key
-  Unique key
-  Unique key disabled
-  User-defined unique key
-  Active trigger
-  Disabled trigger
-  Many to one relation
-  User-defined many to one relation
-  One to many relation
-  User-defined one to many relation
-  Many to many relation
-  User-defined many to many relation
-  One to one relation
-  User-defined one to one relation
-  Input
-  Output
-  Input/Output
-  Uses dependency
-  User-defined uses dependency
-  Used by dependency
-  User-defined used by dependency



## Registration flow:















# 1. User registration




## 1.1. Tables

### 1.1.1. Collection: account (User account details)






#### Columns

Name		Data type	Description / Attributes
 	_id	Id	
	type	String	
	businessDetails	String	
	businessDetails.name	String	
	businessDetails.website	String	
	businessDetails.location	String	
	businessDetails.address	String	
	businessDetails.companySize	Number	
	masteUser	Id	<b>References:</b> users (Users schema)
	created_At	datetime	
	updated_At	datetime	


#### Links to

Table	Join	Title / Name / Description
 users (Users schema)	<b>account</b> .masteUser = users_id	fk_users_account

#### Linked from








Table	Join	Title / Name / Description
 invitation (invitation link data with validity)	<b>account</b> _id = invitationaccount	fk_account_invitation
 license (User or business licenses)	<b>account</b> _id = licenseaccount	fk_account_license
 meetings (All meetings data here)	<b>account</b> _id = meetingsaccount	fk_account_meetings
 payments (all inward and outward payment details)	<b>account</b> _id = payments_id	fk_account_payments
 users (Users schema)	<b>account</b> _id = usersaccounts	fk_account_users

#### Unique keys



Columns	Name / Description
 _id	pk_account_id

### 1.1.2. Collection: license (User or business licenses)



#### Columns

Name		Data type	Description / Attributes
	_id	Id	
	shortId	String	
	user	Id	
	account	Id	<b>References:</b> account (User account details)
	purchaseDate	datetime	
	expiryDate	dateTime	
	invoice	Id	<b>References:</b> invoice (All invoice generated)

#### Links to

Table	Join	Title / Name / Description
 account (User account details)	<b>license</b> account = account_id	fk_account_license
 invoice (All invoice generated)	<b>license</b> invoice = invoice_id	fk_invoice_license

#### Linked from
































Table	Join	Title / Name / Description
 invitation (invitation link data with validity)	<b>license</b> _id = invitationlicenseId	fk_license_invitation
 users (Users schema)	<b>license</b> _id = userslicenses	fk_license_users

#### Unique keys



Columns	Name / Description
 _id	pk_license__id
 shortId	uk_license_shortId

### 1.1.3. Collection: users (Users schema)




#### Columns

	Name	Data type	Description / Attributes
	 _id	Id	Nullable
	status	String	Nullable
	name	String	Nullable
	 email	String	Nullable
	password	String	Nullable
	 phone	String	Nullable
	addresses	Document[]	Nullable
	addresses._id	Id	Nullable
	addresses.name	String	Nullable
	addresses.phone	String	Nullable
	addresses.addressLine1	String	Nullable
	addresses.addressLine2	String	Nullable
	addresses.locality	String	Nullable
	addresses.city	String	Nullable
	addresses.state	String	Nullable
	addresses.postalCode	String	Nullable
	addresses.lat	Double	Nullable
	addresses.lng	Double	Nullable
	createdAt	DateTime	Nullable
	updatedAt	DateTime	Nullable
	__v	Int32	Nullable
	bankAccount	Document	Nullable
	bankAccount.accountNumber	String	Nullable
	bankAccount.IFSC	String	Nullable
	bankAccount.name	String	Nullable
	isActive	Boolean	
	licenses	Id	Nullable References: license (User or business licenses)
	accounts	Id	Nullable References: account (User account details)

#### Links to

	Table	Join	Title / Name / Description
	account (User account details)	<b>users</b> accounts = account_id	fk_account_users
	license (User or business licenses)	<b>users</b> licenses = license_id	fk_license_users

## Linked from

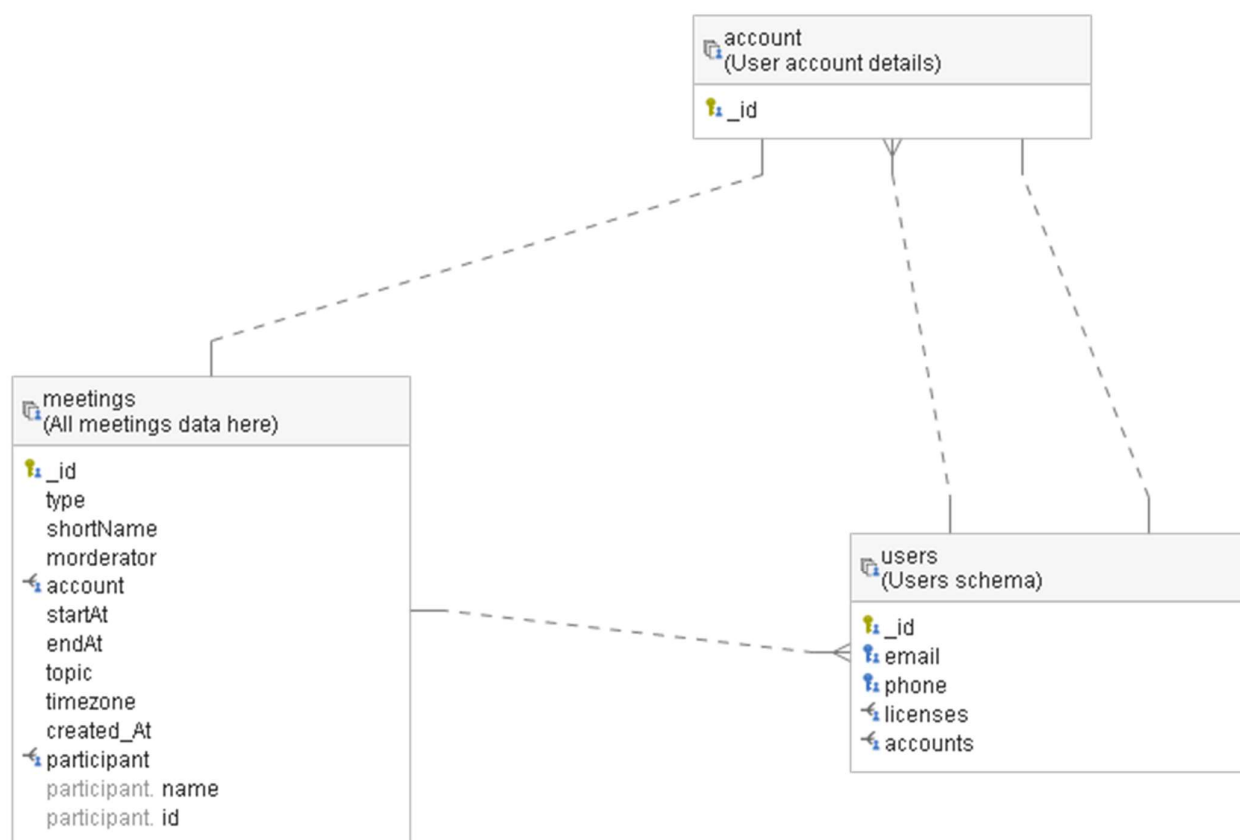
	Table	Join	Title / Name / Description
	account (User account details)	<b>users_id</b> = accountmasterUser	fk_users_account
	invitation (invitation link data with validity)	<b>users_id</b> = invitationby	fk_users_invitation
	meetings (All meetings data here)	<b>users_id</b> = meetingsparticipant	fk_users_meetings

## Unique keys

	Columns	Name / Description
	_id	pk_users__id
	email, phone	uk_users_email_phone



## 2. Meeting scheduler


















## 2.1. Tables

### 2.1.1. Collection: account (User account details)






#### Columns

Name		Data type	Description / Attributes
 	_id	Id	
	type	String	
	businessDetails	String	
	businessDetails.name	String	
	businessDetails.website	String	
	businessDetails.location	String	
	businessDetails.address	String	
	businessDetails.companySize	Number	
	masterUser	Id	<b>References:</b> users (Users schema)
	created_At	datetime	
	updated_At	datetime	


#### Links to

Table	Join	Title / Name / Description
 users (Users schema)	<b>account</b> masterUser = users_id	fk_users_account

#### Linked from














Table	Join	Title / Name / Description
 invitation (invitation link data with validity)	<b>account</b> _id = invitationaccount	fk_account_invitation
 license (User or business licenses)	<b>account</b> _id = licenseaccount	fk_account_license
 meetings (All meetings data here)	<b>account</b> _id = meetingsaccount	fk_account_meetings
 payments (all inward and outward payment details)	<b>account</b> _id = payments_id	fk_account_payments
 users (Users schema)	<b>account</b> _id = usersaccounts	fk_account_users

#### Unique keys



Columns	Name / Description
 _id	pk_account_id

## 2.1.2. Collection: meetings (All meetings data here)


### Columns

Name		Data type	Description / Attributes
	_id	Id	
	type	String	Instant    Scheduled
	shortName	String	
	morderator	String	
	account	Id	<b>References:</b> account (User account details)
	startAt	datetime	
	endAt	datetime	
	topic	String	
	timezone	String	
	created_At	datetime	
	participant		<b>References:</b> users (Users schema)
	participant.name	String	
	participant.id	Id	<b>Nullable</b>

### Links to





























Table	Join	Title / Name / Description
 account (User account details)	<b>meetings</b> account = account_id	fk_account_meetings
 users (Users schema)	<b>meetings</b> participant = users_id	fk_users_meetings

### Unique keys



Columns	Name / Description
 _id	pk_meetings__id

### 2.1.3. Collection: users (Users schema)




#### Columns

	Name	Data type	Description / Attributes
	_id	Id	Nullable
	status	String	Nullable
	name	String	Nullable
	email	String	Nullable
	password	String	Nullable
	phone	String	Nullable
	addresses	Document[]	Nullable
	addresses._id	Id	Nullable
	addresses.name	String	Nullable
	addresses.phone	String	Nullable
	addresses.addressLine1	String	Nullable
	addresses.addressLine2	String	Nullable
	addresses.locality	String	Nullable
	addresses.city	String	Nullable
	addresses.state	String	Nullable
	addresses.postalCode	String	Nullable
	addresses.lat	Double	Nullable
	addresses.lng	Double	Nullable
	createdAt	DateTime	Nullable
	updatedAt	DateTime	Nullable
	__v	Int32	Nullable
	bankAccount	Document	Nullable
	bankAccount.accountNumber	String	Nullable
	bankAccount.IFSC	String	Nullable
	bankAccount.name	String	Nullable
	isActive	Boolean	
	licenses	Id	Nullable References: license (User or business licenses)
	accounts	Id	Nullable References: account (User account details)

#### Links to

	Table	Join	Title / Name / Description
	account (User account details)	<b>users</b> accounts = account_id	fk_account_users
	license (User or business licenses)	<b>users</b> licenses = license_id	fk_license_users

## Linked from

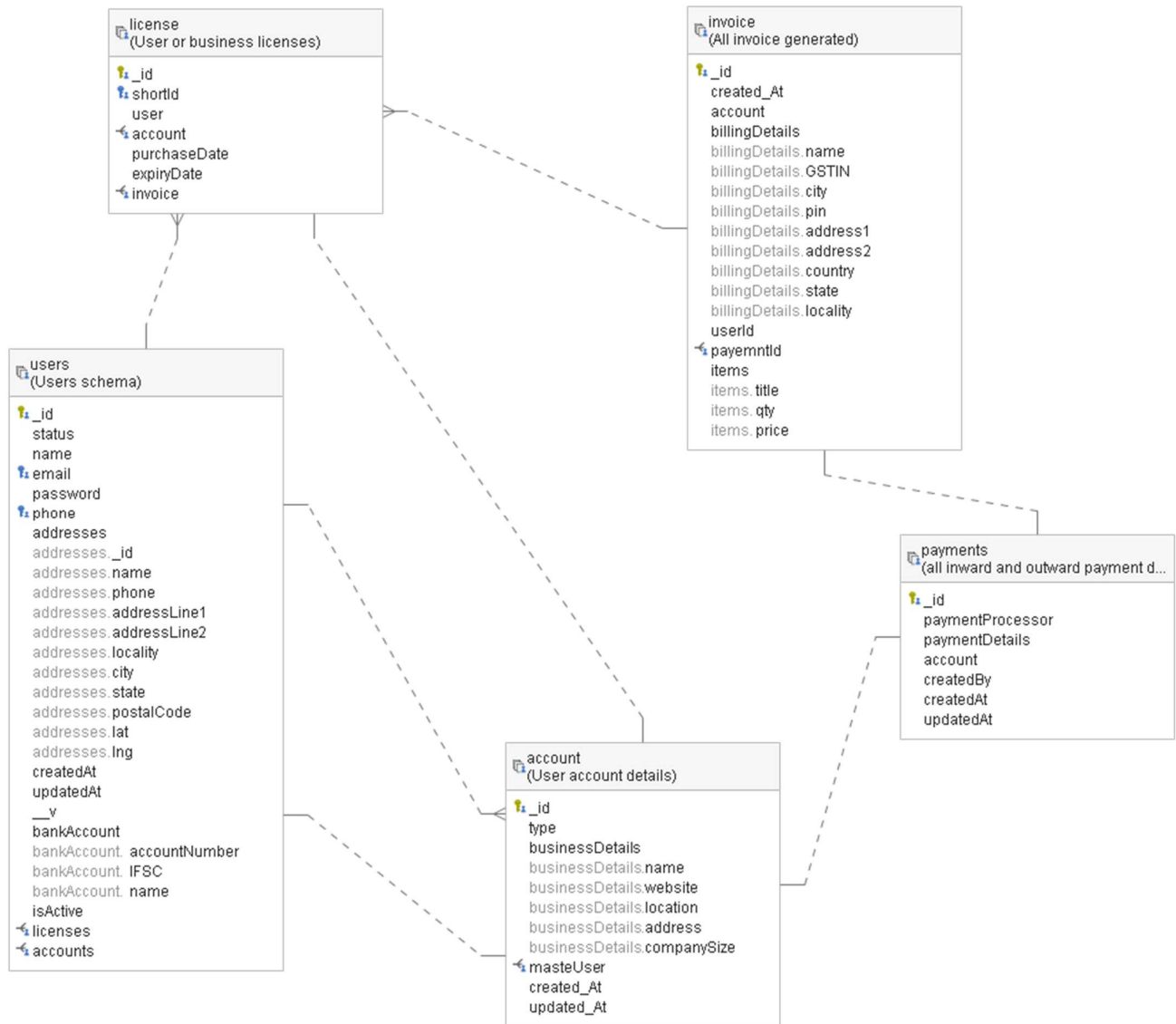
	Table	Join	Title / Name / Description
	account (User account details)	<b>users_id</b> = accountmasterUser	fk_users_account
	invitation (invitation link data with validity)	<b>users_id</b> = invitationby	fk_users_invitation
	meetings (All meetings data here)	<b>users_id</b> = meetingsparticipant	fk_users_meetings

## Unique keys

	Columns	Name / Description
	_id	pk_users__id
	email, phone	uk_users_email_phone















### 3. Subscriptions




## 3.1. Tables

### 3.1.1. Collection: account (User account details)






#### Columns

Name		Data type	Description / Attributes
 	_id	Id	
	type	String	
	businessDetails	String	
	businessDetails.name	String	
	businessDetails.website	String	
	businessDetails.location	String	
	businessDetails.address	String	
	businessDetails.companySize	Number	
	masterUser	Id	<b>References:</b> users (Users schema)
	created_At	datetime	
	updated_At	datetime	


#### Links to

Table	Join	Title / Name / Description
 users (Users schema)	<b>account</b> masterUser = users_id	fk_users_account

#### Linked from




















Table	Join	Title / Name / Description
 invitation (invitation link data with validity)	<b>account</b> _id = invitationaccount	fk_account_invitation
 license (User or business licenses)	<b>account</b> _id = licenseaccount	fk_account_license
 meetings (All meetings data here)	<b>account</b> _id = meetingsaccount	fk_account_meetings
 payments (all inward and outward payment details)	<b>account</b> _id = payments_id	fk_account_payments
 users (Users schema)	<b>account</b> _id = usersaccounts	fk_account_users

#### Unique keys


Columns	Name / Description
 _id	pk_account_id

### 3.1.2. Collection: invoice (All invoice generated)

#### Columns

	Name	Data type	Description / Attributes
	_id	Id	
	created_At	datetime	
	account	Id	
	billingDetails	String	
	billingDetails.name	String	
	billingDetails.GSTIN	String	
	billingDetails.city	String	
	billingDetails.pin	String	
	billingDetails.address1	String	
	billingDetails.address2	String	
	billingDetails.country	String	
	billingDetails.state	String	
	billingDetails.locality	String	
	userId	Id	
	payemntId	Id	<b>References:</b> payments (all inward and outward payment details)
	items	String	
	items.title	String	
	items.qty	int	
	items.price	int	


#### Links to

	Table	Join	Title / Name / Description
	payments (all inward and outward payment details)	<b>invoice</b> payemntId = payments_id	fk_payments_invoice

#### Linked from

	Table	Join	Title / Name / Description
	license (User or business licenses)	<b>invoice</b> _id = licenseinvoice	fk_invoice_license








#### Unique keys

	Columns	Name / Description
	_id	pk_invoice__id





### 3.1.3. Collection: license (User or business licenses)



#### Columns

Name		Data type	Description / Attributes
	_id	Id	
	shortId	String	
	user	Id	
	account	Id	<b>References:</b> account (User account details)
	purchaseDate	datetime	
	expiryDate	dateTime	
	invoice	Id	<b>References:</b> invoice (All invoice generated)

#### Links to

Table	Join	Title / Name / Description
 account (User account details)	<b>license</b> account = account_id	fk_account_license
 invoice (All invoice generated)	<b>license</b> invoice = invoice_id	fk_invoice_license

#### Linked from








Table	Join	Title / Name / Description
 invitation (invitation link data with validity)	<b>license</b> _id = invitationlicenseId	fk_license_invitation
 users (Users schema)	<b>license</b> _id = userslicenses	fk_license_users

#### Unique keys


Columns	Name / Description
 _id	pk_license__id
 shortId	uk_license_shortId

### 3.1.4. Collection: payments (all inward and outward payment details)


#### Columns

Name		Data type	Description / Attributes
	_id		<b>References:</b> account (User account details)
	paymentProcessor		
	paymentDetails		
	account		
	createdBy		
	createdAt		
	updatedAt		

#### Links to

Table	Join	Title / Name / Description
 account (User account details)	<b>payments_id</b> = account_id	fk_account_payments

#### Linked from
































Table	Join	Title / Name / Description
 invoice (All invoice generated)	<b>payments_id</b> = invoicepayemntId	fk_payments_invoice

#### Unique keys



Columns	Name / Description
 _id	pk_payments__id

### 3.1.5. Collection: users (Users schema)




#### Columns

	Name	Data type	Description / Attributes
	 _id	Id	Nullable
	status	String	Nullable
	name	String	Nullable
	 email	String	Nullable
	password	String	Nullable
	 phone	String	Nullable
	addresses	Document[]	Nullable
	addresses._id	Id	Nullable
	addresses.name	String	Nullable
	addresses.phone	String	Nullable
	addresses.addressLine1	String	Nullable
	addresses.addressLine2	String	Nullable
	addresses.locality	String	Nullable
	addresses.city	String	Nullable
	addresses.state	String	Nullable
	addresses.postalCode	String	Nullable
	addresses.lat	Double	Nullable
	addresses.lng	Double	Nullable
	createdAt	DateTime	Nullable
	updatedAt	DateTime	Nullable
	__v	Int32	Nullable
	bankAccount	Document	Nullable
	bankAccount.accountNumber	String	Nullable
	bankAccount.IFSC	String	Nullable
	bankAccount.name	String	Nullable
	isActive	Boolean	
	licenses	Id	Nullable References: license (User or business licenses)
	accounts	Id	Nullable References: account (User account details)

#### Links to

	Table	Join	Title / Name / Description
	account (User account details)	<b>users</b> accounts = account_id	fk_account_users
	license (User or business licenses)	<b>users</b> licenses = license_id	fk_license_users

## Linked from

	Table	Join	Title / Name / Description
	account (User account details)	<b>users_id</b> = accountmasterUser	fk_users_account
	invitation (invitation link data with validity)	<b>users_id</b> = invitationby	fk_users_invitation
	meetings (All meetings data here)	<b>users_id</b> = meetingsparticipant	fk_users_meetings

## Unique keys

	Columns	Name / Description
	_id	pk_users__id
	email, phone	uk_users_email_phone



## 4. user dashboard













## 5. Calendar




## 5.1. Tables

### 5.1.1. Collection: account (User account details)






#### Columns

Name		Data type	Description / Attributes
 	_id	Id	
	type	String	
	businessDetails	String	
	businessDetails.name	String	
	businessDetails.website	String	
	businessDetails.location	String	
	businessDetails.address	String	
	businessDetails.companySize	Number	
	masterUser	Id	<b>References:</b> users (Users schema)
	created_At	datetime	
	updated_At	datetime	


#### Links to

Table	Join	Title / Name / Description
 users (Users schema)	<b>account</b> masterUser = users_id	fk_users_account

#### Linked from

Table	Join	Title / Name / Description
 invitation (invitation link data with validity)	<b>account</b> _id = invitationaccount	fk_account_invitation
 license (User or business licenses)	<b>account</b> _id = licenseaccount	fk_account_license
 meetings (All meetings data here)	<b>account</b> _id = meetingsaccount	fk_account_meetings
 payments (all inward and outward payment details)	<b>account</b> _id = payments_id	fk_account_payments
 users (Users schema)	<b>account</b> _id = usersaccounts	fk_account_users














#### Unique keys

Columns	Name / Description
 _id	pk_account_id





### 5.1.2. Collection: meetings (All meetings data here)

#### Columns

Name		Data type	Description / Attributes
	_id	Id	
	type	String	Instant    Scheduled
	shortName	String	
	morderator	String	
	account	Id	<b>References:</b> account (User account details)
	startAt	datetime	
	endAt	datetime	
	topic	String	
	timezone	String	
	created_At	datetime	
	participant		<b>References:</b> users (Users schema)
	participant.name	String	
	participant.id	Id	<b>Nullable</b>

#### Links to





























Table	Join	Title / Name / Description
 account (User account details)	<b>meetings</b> account = account_id	fk_account_meetings
 users (Users schema)	<b>meetings</b> participant = users_id	fk_users_meetings

#### Unique keys



Columns	Name / Description
 _id	pk_meetings__id

### 5.1.3. Collection: users (Users schema)




#### Columns

	Name	Data type	Description / Attributes
	_id	Id	Nullable
	status	String	Nullable
	name	String	Nullable
	email	String	Nullable
	password	String	Nullable
	phone	String	Nullable
	addresses	Document[]	Nullable
	addresses._id	Id	Nullable
	addresses.name	String	Nullable
	addresses.phone	String	Nullable
	addresses.addressLine1	String	Nullable
	addresses.addressLine2	String	Nullable
	addresses.locality	String	Nullable
	addresses.city	String	Nullable
	addresses.state	String	Nullable
	addresses.postalCode	String	Nullable
	addresses.lat	Double	Nullable
	addresses.lng	Double	Nullable
	createdAt	DateTime	Nullable
	updatedAt	DateTime	Nullable
	__v	Int32	Nullable
	bankAccount	Document	Nullable
	bankAccount.accountNumber	String	Nullable
	bankAccount.IFSC	String	Nullable
	bankAccount.name	String	Nullable
	isActive	Boolean	
	licenses	Id	Nullable References: license (User or business licenses)
	accounts	Id	Nullable References: account (User account details)

#### Links to

	Table	Join	Title / Name / Description
	account (User account details)	<b>users</b> accounts = account_id	fk_account_users
	license (User or business licenses)	<b>users</b> licenses = license_id	fk_license_users

## Linked from

	Table	Join	Title / Name / Description
	account (User account details)	<b>users_id</b> = accountmasterUser	fk_users_account
	invitation (invitation link data with validity)	<b>users_id</b> = invitationby	fk_users_invitation
	meetings (All meetings data here)	<b>users_id</b> = meetingsparticipant	fk_users_meetings

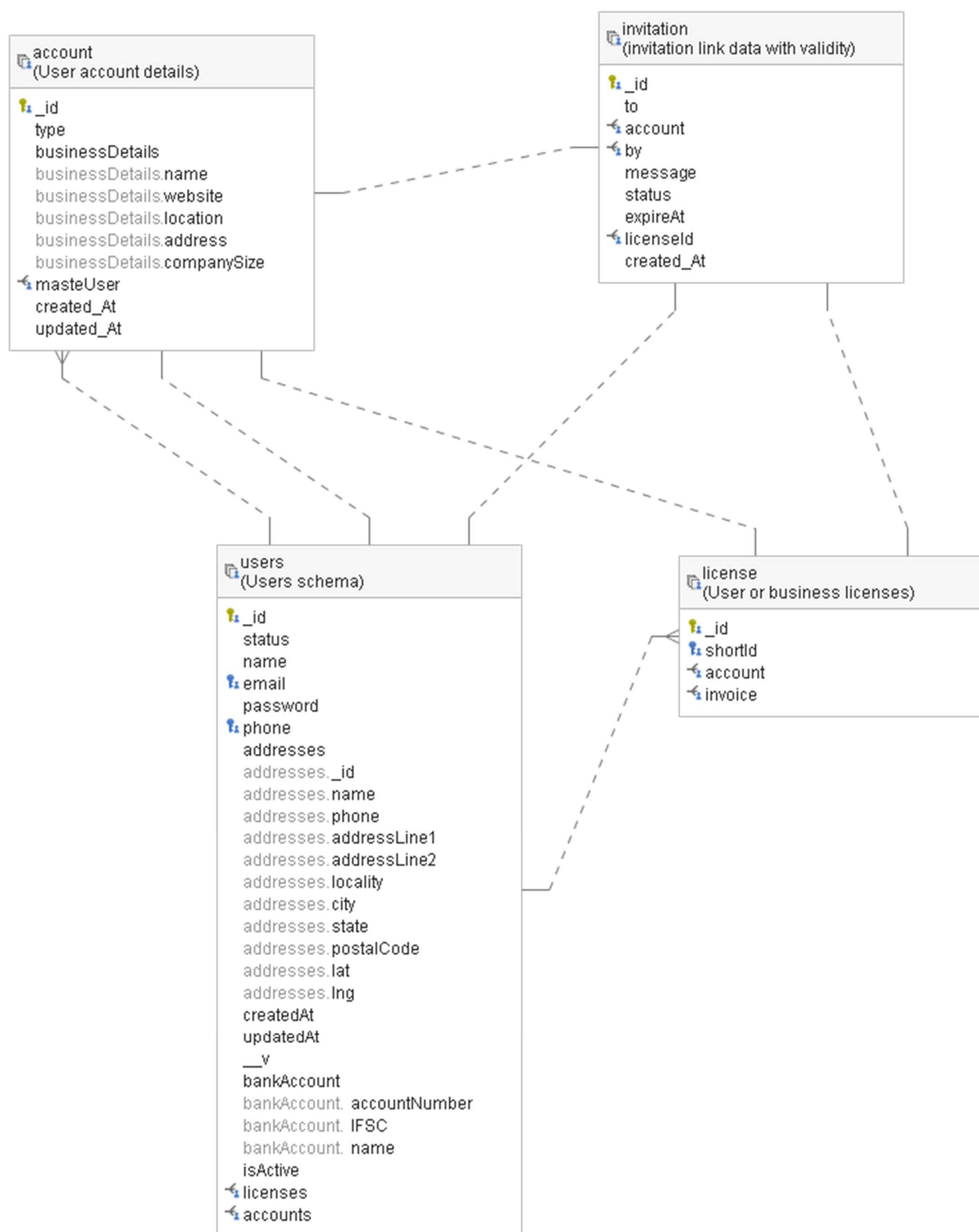
## Unique keys

	Columns	Name / Description
	_id	pk_users__id
	email, phone	uk_users_email_phone



## 6. Customer Admin Panel













## 7. User Invitation




## 7.1. Tables

### 7.1.1. Collection: account (User account details)






#### Columns

Name		Data type	Description / Attributes
 	_id	Id	
	type	String	
	businessDetails	String	
	businessDetails.name	String	
	businessDetails.website	String	
	businessDetails.location	String	
	businessDetails.address	String	
	businessDetails.companySize	Number	
	masterUser	Id	<b>References:</b> users (Users schema)
	created_At	datetime	
	updated_At	datetime	


#### Links to

Table	Join	Title / Name / Description
 users (Users schema)	<b>account</b> masterUser = users_id	fk_users_account

#### Linked from










Table	Join	Title / Name / Description
 invitation (invitation link data with validity)	<b>account</b> _id = invitationaccount	fk_account_invitation
 license (User or business licenses)	<b>account</b> _id = licenseaccount	fk_account_license
 meetings (All meetings data here)	<b>account</b> _id = meetingsaccount	fk_account_meetings
 payments (all inward and outward payment details)	<b>account</b> _id = payments_id	fk_account_payments
 users (Users schema)	<b>account</b> _id = usersaccounts	fk_account_users

#### Unique keys




Columns	Name / Description
 _id	pk_account_id

### 7.1.2. Collection: invitation (invitation link data with validity)


#### Columns

Name		Data type	Description / Attributes
	_id	Id	
	to	Id	
	account	Id	<b>References:</b> account (User account details)
	by	Id	<b>References:</b> users (Users schema)
	message	String	
	status	String	
	expireAt	datetime	
	licenseId	Id	<b>References:</b> license (User or business licenses)
	created_At	datetime	

#### Links to

Table	Join	Title / Name / Description
 account (User account details)	<b>invitation</b> account = account_id	fk_account_invitation
 license (User or business licenses)	<b>invitation</b> licenseId = license_id	fk_license_invitation
 users (Users schema)	<b>invitation</b> by = users_id	fk_users_invitation








#### Unique keys

Columns	Name / Description
 _id	pk_invitation_id





### 7.1.3. Collection: license (User or business licenses)



#### Columns

Name		Data type	Description / Attributes
	_id	Id	
	shortId	String	
	user	Id	
	account	Id	<b>References:</b> account (User account details)
	purchaseDate	datetime	
	expiryDate	dateTime	
	invoice	Id	<b>References:</b> invoice (All invoice generated)

#### Links to

Table	Join	Title / Name / Description
 account (User account details)	<b>license</b> account = account_id	fk_account_license
 invoice (All invoice generated)	<b>license</b> invoice = invoice_id	fk_invoice_license

#### Linked from





























Table	Join	Title / Name / Description
 invitation (invitation link data with validity)	<b>license</b> _id = invitationlicenseId	fk_license_invitation
 users (Users schema)	<b>license</b> _id = userslicenses	fk_license_users

#### Unique keys



Columns	Name / Description
 _id	pk_license__id
 shortId	uk_license_shortId

### 7.1.4. Collection: users (Users schema)




#### Columns

	Name	Data type	Description / Attributes
	 _id	Id	Nullable
	status	String	Nullable
	name	String	Nullable
	 email	String	Nullable
	password	String	Nullable
	 phone	String	Nullable
	addresses	ocument[]	Nullable
	addresses._id	Id	Nullable
	addresses.name	String	Nullable
	addresses.phone	String	Nullable
	addresses.addressLine1	String	Nullable
	addresses.addressLine2	String	Nullable
	addresses.locality	String	Nullable
	addresses.city	String	Nullable
	addresses.state	String	Nullable
	addresses.postalCode	String	Nullable
	addresses.lat	Double	Nullable
	addresses.lng	Double	Nullable
	createdAt	DateTime	Nullable
	updatedAt	DateTime	Nullable
	__v	Int32	Nullable
	bankAccount	Document	Nullable
	bankAccount.accountNumber	String	Nullable
	bankAccount.IFSC	String	Nullable
	bankAccount.name	String	Nullable
	isActive	Boolean	
	licenses	Id	Nullable References: license (User or business licenses)
	accounts	Id	Nullable References: account (User account details)

#### Links to

Table	Join	Title / Name / Description
 account (User account details)	<b>users</b> accounts = account_id	fk_account_users
 license (User or business licenses)	<b>users</b> licenses = license_id	fk_license_users

## Linked from

	Table	Join	Title / Name / Description
	account (User account details)	<b>users_id</b> = accountmasterUser	fk_users_account
	invitation (invitation link data with validity)	<b>users_id</b> = invitationby	fk_users_invitation
	meetings (All meetings data here)	<b>users_id</b> = meetingsparticipant	fk_users_meetings

## Unique keys

	Columns	Name / Description
	_id	pk_users__id
	email, phone	uk_users_email_phone



## 8. Other

### 8.1. Tables

#### 8.1.1. Collection: staff (all staff data with roles her

# Third Party API and Integrations:

## Ip to Country:

There are a lot of APIs available, though we'll need to test on the scale for rate limits. One such API is <http://www.geoplugin.net/json.gp?ip={ip}> and it provides sufficient details.

## Email:

Any SMTP credentials will work but for minimum failure and fast delivery, something like **sendgrid** or **AWS Simple Email Service** is good, it can be easily integrated.

## SMS:

**Twilio** has SDK for Node, and can be easily integrated, but we're open to other options as well.

## Payment Gateway:

### For India:

#### Razorpay:

Razorpay has a nice payment flow, where every payment can be associated to an order, to avoid double payment or similar scenarios, a nice dashboard to double check, a good SDK to integrate and very clear API documentation.

### For countries other than India:

#### Stripe:

**Stripe** has a very clear documentation and SDK for React and Node, with clear examples.

