**Planzz.com**

System description & technical requirements

June 2015

1. Intro

Planzz is a system for managing the workflow of large construction projects with repetitive multiple units such as offices and apartment buildings.

The main entities managed by the system are:

* 1. Units being built. Described by their location in the project.
  2. Flow of Activities required to be completed in every unit.
  3. Relations between the activities – which activities depends on the completion of the other.
  4. Resources – the teams and professionals that are required to perform each activity.

The planzz system allow project managers and resource team leaders to get a clear snapshot status of the project, see clearly what is the critical path and what activity must be done next and also perform advanced resource planning and project timeline.

1. General requirements
   1. The system will be implemented as a web application using the MEAN technology stack: node.js in server side, angular.js on client and either mongoDb or MySQL for the database .
   2. The system will be designed as a single page application with server side providing only a RESTful interface to provide data and algorithm results.
   3. The client will be responsive and will work seamlessly on desktops, tablets and smartphones. It will be implemented using the following twitter bootstrap based toolset: <https://wrapbootstrap.com/theme/angle-bootstrap-admin-app-angularjs-WB04HF123>. A license will be provided.
   4. All source code will be managed in our git repository and will be pushed daily.
   5. The system will allow internationalization-localization (multi language support) to allow easy translation. And also right to left languages support.
   6. The system will implement a payments system using stripe.js. Payment is done per project, allowing the user who created the project to pay using a credit card. The same screen also allows cancelling of the monthly payment. This causes project freeze – no access for any activity is allowed on the project until payments are set.
   7. The project actions and status will be saved to the database with every change submitted by the user.
   8. Every day at 00:00 a snapshot of each project status will be saved to disk for backup and legal inquiries.
   9. every user action will be saved to a log file containing all the history of all user actions. There will be a unified format for this actions log.

1. Project definition screen
2. Create / Edit /delete a **project** with a name property set by the user.

**all the following stages and inputs in this screen are done within the context of a single project.**

1. Create/edit/delete **structures** with the following properties:
   1. Structure name.
   2. Number of **floors** per structure.
   3. Number of units in each floor.
2. For each **Unit**:
   1. Name (supply an automated name as default built from structure id – floor id – running id (01-05-01, 01-05-02 etc.)
   2. **Unit Type** from an options bank (starts empty and if user adds something not in the bank a small confirmation form opens to make sure it's not a typo).
3. Since real projects has hundreds and even thousands of Units, it is highly important to allow excel like ability to quickly duplicate floors and Units with automatic numbers so users won't have to type all manually.
4. Each project has a payments tab allowing the project admin (the user who created it) to enter payments using stripe.js

|  |  |  |  |
| --- | --- | --- | --- |
| Structure | Floor | Unit Number | Unit Type |
| Structure A05 | Floor 01 | A05-01-01 | A |
| A05-01-02 | A |
| A05-01-03 | B |
| Floor 02 | A05-02-01 | A |
| A05-02-02 | C |
| A05-02-03 | C |
| Floor 03 | A05-03-01 | B |
| A05-03-02 | B |
| Structure B12 | Floor 01 | B12-01-01 | D |
| B12-01-02 | D |
| B12-01-03 | D |
| B12-01-04 | C |
| Floor 02 | B12-02-01 | A |
| B12-02-02 | A |
| B12-02-03 | A |
| B12-02-04 | A |

Figure 1 – the project structure screen. Might be designed differently by designer.(design notes: consider separating the structures and changing to a different concept other than a table look as long as data input is fast and intuitive).

1. Resources screen and user permissions screen
   1. The system will implement an email-password login system using passport.js. passwords will be saved encrypted in the database. The system will allow password reset by email.
   2. Each project has a single user who created the project. This is the **project admin**. He is the only one allowed admin access to the project (Project definition & payments screen)
   3. The system will allow anyone to register with a valid email.
   4. Access to existing project is only allowed by invitation that is sent by the admin in the resources page.
   5. The resources screen is defined per project and allows the project admin to feed a list of resources – the companies and people in charge of a specific activity with the following details:
      1. Resource name (From a bank – like previous banks starts empty and grows with notice)
      2. Company name
      3. Contact Person name
      4. Email

Once the user created a new contact that is not yet in the system, an invitation email is sent. Existing users that received a new role in the project also receive a notice mail.

4.7 The resource screen will allow sorting of the table and search.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Resource** | **Company** | **Contact name** | **Contact email** |
| 1 | Cleaners | The Cleaners Inc. | Bob Marley | [bob@gmail.com](mailto:bob@gmail.com) |
| 2 | Electricians | The Electrons Inc. | Bob Dylan | [dylan@yahoo.com](mailto:dylan@yahoo.com) |
| 3 | Plumbers | The Plumbers Inc. | Bruce Springsteen | [bruce@abc.com](mailto:bruce@abc.com) |
| 4 | Markers | The Best Markers | Elvis Presley | [elvis@markers.com](mailto:elvis@markers.com) |

**Important: Notice that currently the system will only support one company per resource but multiple resources per company**.

Figure 2 – the resource input screen as a simple table – Possibly will designed better.

1. Activities screen
   1. All of the activities related data is defined separately per project.
   2. The screen allow Create / edit the project Flow. Allows importing of existing flows from other projects of the same user.
   3. In each flow add/remove/edit the list of activities from a bank of the existing activities (starts empty and when a new activity that doesn’t exists is added a form pops up validating it).
   4. An Activity is comprised of the following data:
      1. Activity Name
      2. Pre-Activity – a dropbox of all other activities in this flow for user to choose which activity is the previous one required to be finished in order to start this one. This field can be left empty for activities with no prerequisites.
      3. The resource type responsible for performing this Activity.

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Activity** | **Pre-Activity** | **Resource** |
| 1 | Cleaning | - | Cleaning |
| 2 | Marking | - | Markers |
| 3 | Plumbing | 2-Marking | Plumbers |
| 4 | Plumbing test | 3-Plumbing | Plumbers |
| 5 | Electric infra | 4-Plumbing test | Electricians |
| 6 | Electric Infra test | 5-Electric infra | Electricians |
|  |  |  |  |

* 1. Once the Activities list is entered The Activities relevancy table is available for use. In this table the user can choose which of the activities is not relevant to specific **Unit types**. All starts as relevant and user deselects the combination not relevant.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Type A | Type B | Type C |
| Cleaning | v | v | x |
| Marking | v | v | x |
| Plumbing | v | v | v |
| Plumbing test | v | x | x |
| Electric infra | v | v | v |
| Electric Infra test | v | v | v |
| Floor application | v | x | x |

1. Main project screen
   1. After the user created the project and entered all the flows and resources required he clicks on "Show Project Status" and the following screen is presented.
   2. This screen allows a snapshot of the project status in a single screen.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Activity | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | Cleaning | A05-01-01 | A05-01-02 | A05-01-03 | A05-02-01 | A05-02-02 | A05-02-03 | A05-03-01 | A05-03-02 |
| 2 | Marking |  | A05-01-01 | A05-01-02 | A05-01-03 | A05-02-01 | A05-02-02 | A05-02-03 | A05-03-01 |
| 3 | Plumbing |  |  | A05-01-01 | A05-01-02 | A05-01-03 | A05-02-01 | A05-02-02 | A05-02-03 |
| 4 | Plumbing test |  |  |  | A05-01-01 | A05-01-02 | A05-01-03 | A05-02-01 | A05-02-02 |
| 5 | Electric infra |  |  |  |  | A05-01-01 | A05-01-02 | A05-01-03 | A05-02-01 |
| 6 | Electric Infra test |  |  |  |  |  | A05-01-01 | A05-01-02 | A05-01-03 |
| 7 | Floor application |  |  |  |  |  |  | A05-01-01 | A05-01-02 |

* 1. The colors of the cells indicate its reported status:

|  |
| --- |
| Fully Done or not relevant to this apartment |
| Almost done next activity can begin |
| Activity started |

* 1. The rows shows the activity flow defined above for the project.
  2. The columns represents a time slice in the project numbered from 1 to the required number of time slices.
  3. The cells contain the Unit Ids orders by their defined order in each row and shifts forward in time between lines because each activity that depends on previous one can't start in a unit before the previous one finished.

1. Progress Reporting screen
   1. This is the screen where the resource managers can report work done in the project. When a user logs in he can either go to "My Projects" for projects he is the admin for or "Reported Projects" for projects he is a resource in.
   2. Once he chooses "Reported Projects" and chooses a specific project the report screen for this project is opened for him – only for the resources he is assigned to. So if Bob Marley is assigned as a cleaner in our project he will only be able to report clean activity in this project.
   3. The user will visually see in which of the Units his activities are next in flow (will see the full project screen and only his relevant units will be colored – designer decision).
   4. Once he chooses a colored Unit a floating form opens that allows him to choose the activity if there is more than one and one of the following reports:
2. Activity not required in this unit
3. Activity started
4. Almost finished – next activity can start
5. Full completion

Once the user selected the report and clicked "Report" an alert shows saying "Reports can't be changed – are you sure". Once the user chose Yes - This report cannot be changed anymore and is saved.

If the user chose to report on a unit not colored for him – a floating form pops asking:

You are trying to report a unit which you are not assigned to. Cancel to abort or choose the reason for the report :

1. I can start work in this unit – my previous forgot to report finished

2. I can start work in this unit – my previous still working but does not interrupt me.

1. Reports screen

Each resource team leader needs to be able to see his own work status and history.

The screen will show to the user the project main screen for his relevant data:

* apartments he finished
* apartments he is in progress
* apartments waiting for his work.

In addition this screen will have a log table for this resource team showing his actions: each activity report with its date-time with search option.

1. System admin page
   1. Allows defining default price per Unit in USD.
   2. Allows defining the default free trial period.
   3. Allows changing free periods per project
   4. Allows changing price per Unit per specific projects.