FlowMarkt UI Design and Prototyping Report

Project Title

1. Introduction

- FlowMarkt lowers transaction costs of recycling: time spent presenting and finding items, negotiating conditions as well as delivering goods. Users are geolocated, and near proximity buyers are preferred. FlowMarkt integrates to social media and digital communication tools to bring together remote users. FlowMarkt tracks “strength of trust”, which is calculated from past experiences, proximity and connection strenth between parties. Trust between parties is interpreted as “social capital” which automatically gives priority when FlowMarkt recommends further sale or exchange possibilities.

2. User Interface Design and Prototype

- Users can be categorized to groups or parties of sales transaction like
  - “Selling” or “Giving” party or “Presenter” of items
  - “Buying” or “Finding” party or “Browser” of items
  - Real users are often in both roles during same usage session
  - User “prototypes” can be effectively described using personas, which makes it clear that punk rock collector is very different type of user than mom searching clothes for kids https://www.smashingmagazine.com/2014/08/a-closer-look-at-personas-part-1/
- User interface definition should start small. One way is developing simple wireframe mockups.
- Mockups present MVP (minimal viable product) and are enriched when there's more expericence of user's needs. Users' needs are collected thru interviews and demos.
- Mockups can be done with plantuml's salt dsl (domain specific language) http://plantuml.com/salt and generated from text online ww.plantuml.com/plantuml/.
- PlantUml is based on GraphViz http://www.graphviz.org/
- Mockups are done to be thrown away early as during the process there comes opportunities to detail them and do actionable demos using tools like Ionic Creator http://ionic.io/products/creator, Balsamiq https://balsamiq.com/products/mockups/ or alike
- Certainty of goal is incrementally distilled to solution – at the end it shouldn't be just a rehersal for nothing, but useful app for real users, or minimum amout of wasted time.

2.1 Login

First executable version of demo can be implemented using PostgRest http://postgrest.com. PostgRest is api for Postgres SQL database https://www.postgresql.org/. Database schema can be written directly with SQL as DSL http://www.andrejkoelewijn.com/blog/2008/10/27/sql-is-a-dsl/

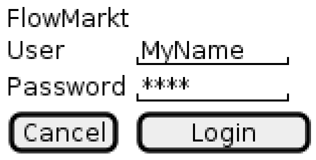
After initial setup of PostgRest http call to authenticate user is pretty simple http://postgrest.com/examples/users/

```
POST /rpc/login
```

```
{ "email": "foo@bar.com", "pass": "foobar" }
```

Fields needed for authentication call are implemented at login form. Form visualization and Salt DSL are here to be seen side to side.

```
@startsalt
{
FlowMarkt

  User     | "MyName   "
  Password | "****     "
  [Cancel] | [  Login   ]

}
@endsalt
```



FlowMarkt

User	MyName
Password	****

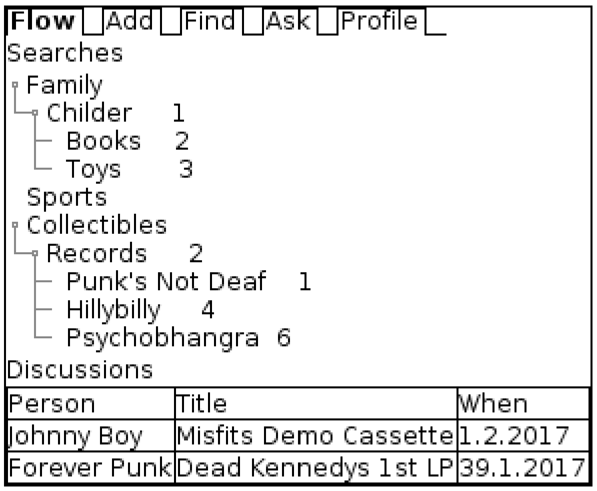
After login application is secured using JWT https://jwt.io/

2.2 Main Dashboard view (Flow)

Opening screen is Dashboard and presents summary of running user interactions and results of saved searches https://en.wikipedia.org/wiki/Dashboard_%28business%29. During development dashboard will be exteded with information that summarizes active state of users operations.

For search results we can use treelike structure which contains relevant new results of saved searches. Here tree is shown as hierarchy, but it could be also collapsible structure which has clear root like here https://bl.ocks.org/mbostock/4339083. Discussions are shown as list.

```
@startsalt
{+
{/ <b>Flow | Add | Find | Ask | Profile }

Searches
{T
 + Family
 ++ Childer     1
 +++ Books    2
 +++ Toys       3
 + Sports
 + Collectibles
 ++ Records     2
 +++ Punk's Not Deaf    1
 +++ Hillybilly     4
 +++ Psychobhangra  6
}

Discussions
{#
Person | Title | When
Johnny Boy | Misfits Demo Cassette | 1.2.2017
Forever Punk | Dead Kennedys 1st LP | 39.1.2017
}

}
@endsalt
```



Searches

```
Family
  Childer    1
    Books    2
    Toys     3
Sports
Collectibles
  Records    2
    Punk's Not Deaf    1
    Hillybilly    4
    Psychobhangra  6
```

Discussions

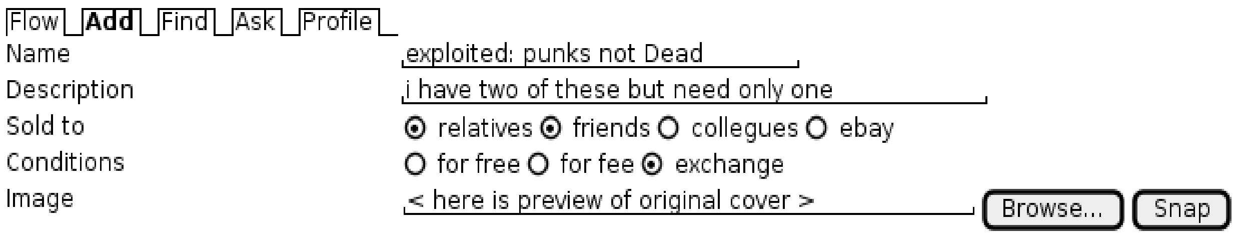
Person	Title	When
Johnny Boy	Misfits Demo Cassette	1.2.2017
Forever Punk	Dead Kennedys 1st LP	39.1.2017

Dashboard is likely to be highly configurable thru settings given using profile. How these settings are given and what default settings application has will be found during UX sessions.

2.3 Posting items (Add)

During presentation of goods seller gives item simple name, short description, target group and conditions. Picture can be found using gallery or taken directly.

```
@startsalt
{
{/ Flow | <b>Add | Find | Ask | Profile }

Name         | " exploited: punks not Dead "
Description  | " i have two of these but need only one  "

Sold to   | { (X) relatives | (X) friends | () collegues | () ebay }
Conditions   | { () for free | () for fee | (X) exchange  }

Image  | { " < here is preview of original cover > " | [Browse...] | [ Snap ] }
}
@endsalt
```



Flow | **Add** | Find | Ask | Profile

Name	exploited: punks not Dead
Description	i have two of these but need only one
Sold to	[✓ relatives] [✓ friends] collegues ebay
Conditions	for free for fee [✓ exchange]
Image	< here is preview of original cover > Browse... Snap

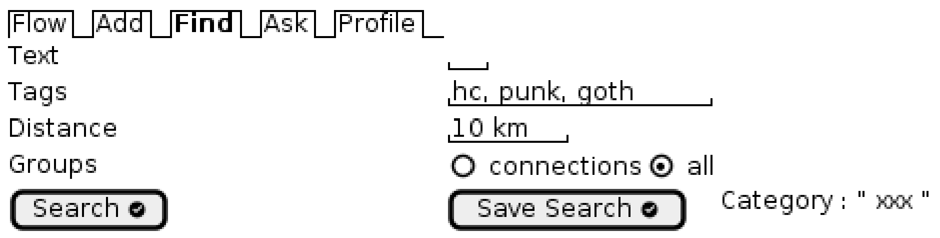
It is likely that adding item needs several steps as all information might not fit to single screen.

2.4 Searching items (Find)

Finding items allow executinion of ad hoc queries and saving queries for later use. Search is targeted to given set of sellers within defined proximity.

```
@startsalt
{
{/ Flow | Add | <b>Find | Ask | Profile }

Text           | "  "
Tags           | " hc, punk, goth "

Distance   | " 10 km "
Groups     | { () connections | (X) all }

[Search <&circle-check>] | [Save Search <&circle-check>] | Category : “ xxx “

}
```

```
@endsalt
```



Text	
Tags	hc, punk, goth
Distance	10 km
Groups	connections [✓ all]

Category : " xxx "

Saved queries can be edited and further configured under profile. Saved searches are grouped under single category. Categories can be organized in parent-child relationship under profiles.

2.4 Q&A, Chatting and asking details (Ask)

Not detailed. Long running negotiation process allows seller to answer questions, buyer to ask more detail questions, give propositions and agree on conditions.

2.5 Changing settings, redefining searches, grouping connections, etc. (Profile)

Not detailed. All user profile stuff and classifications, which make user experience truly personalized.

Profile could contain for example
- Name, location, picture, etc. personal info
- Categories for search, tags, etc. classifying structures
- Connections to backend systems with configuration
- Friends, Collegues, and other groups of people
- History of discussions and sales transactions

3. Navigation Structure

There's only very few (5) main menu items: Flow, Add, Find, Ask, Profile

From these menus users start different flows:

- sellers basic sales flow starts thru “add” : present item with detail information and picture.

- byuers search and buy flow starts thru “find” or “flow” : browse results or saved search or execute search, select item, go on with negotiation if needed.

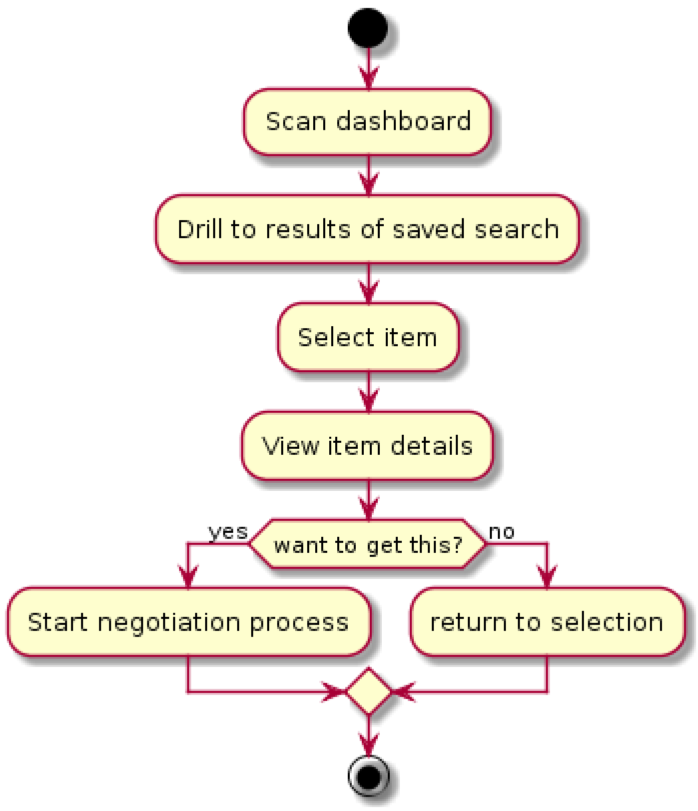
- sellers negotiation flow starts from “ask” or “flow”: see questions, answer, give offers, agree.

There will be many more flows, which link to goals of personas (punk collector “Johnny Boy” wants to find exchange partners for old classics, and for him we might describe “collector” flow)

Flows can be presented using UML notation using PlantUml's activity diagram http://plantuml.com/activity-diagram-beta

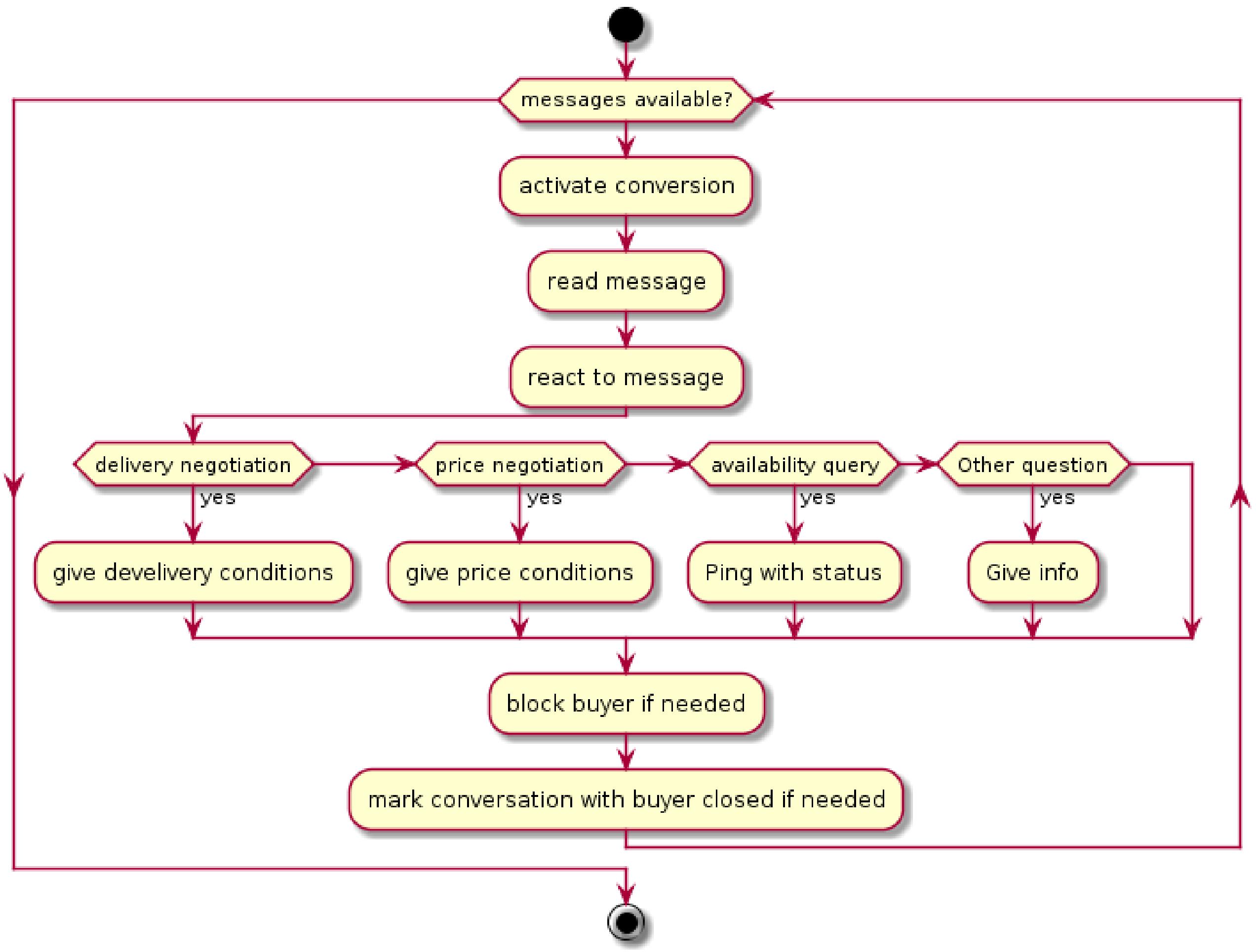
Under is simple presentation of buyers search & buy flow without longrunning negotiation flow, which is described separately.

```
@startuml
start
:Scan dashboard;
:Drill to results of saved search;
:Select item;
:View item details;
if (want to get this?) then (yes)
   :Start negotiation process;
else (no)
  :return to selection;
endif
stop
@enduml
```



Negotiation process needs to be defined from perspective of seller and buyer separately. Here's simplified sellers negotiation process.

```
@startuml
start

while (messages available?)
  :activate conversion;
  :read message;
  :react to message;

if (delivery negotiation) then (yes)
  :give develivery conditions;
elseif (price negotiation) then (yes)
  :give price conditions;
elseif (availability query) then (yes)
  :Ping with status;
elseif (Other question) then (yes)
  :Give info;
endif

  :block buyer if needed;
  :mark conversation with buyer closed if needed;
endwhile

stop
@enduml
```

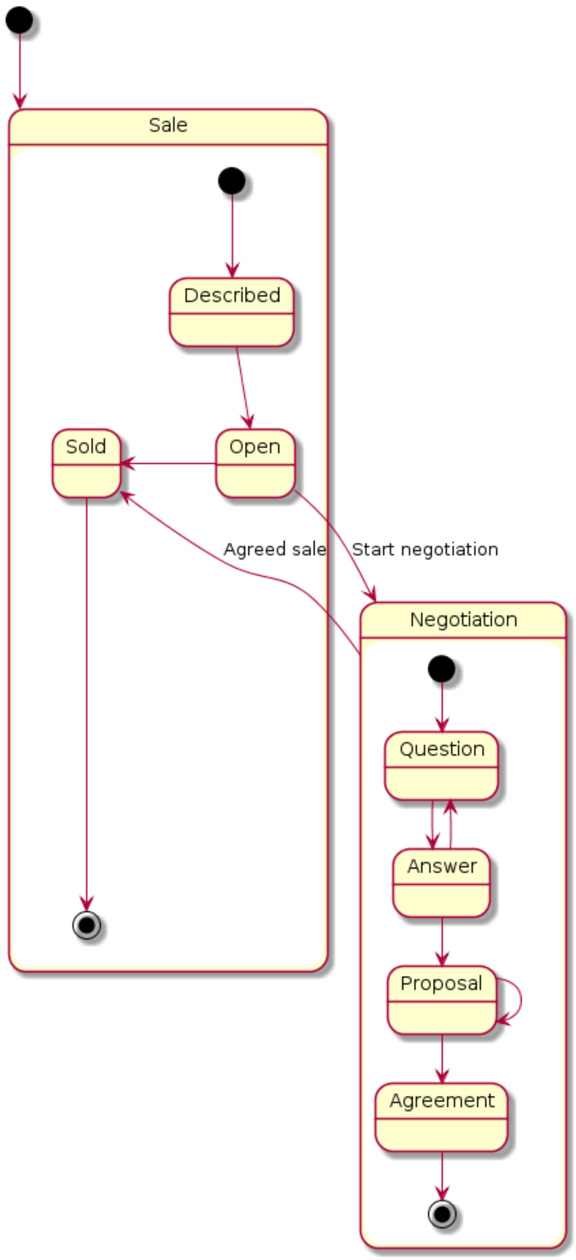


Coordination between flows and asyncronous comminication during long running process could be captured using BPMN (Business Model Notation). Here's simple example https://www.businessprocessincubator.com/content/incident-management-as-detailed-collaboration/

Note that both sales transaction and negotiation could be presented also using state change diagram to illustrate how negotiation can only start when sales transaction is in state of “open for offers”.

State diagrams can be described using PlantUml's state diagram http://plantuml.com/state-diagram

```
@startuml
[*] --> Sale

state Sale {
  [*] --> Described
  Described --> Open
  Open --> Negotiation : Start negotiation
  Negotiation --> Sold : Agreed sale
  Open -> Sold
  Sold --> [*]
}

state Negotiation {
  [*] --> Question
  Question --> Answer
  Answer --> Question
  Answer --> Proposal
  Proposal --> Proposal
  Proposal --> Agreement
  Agreement --> [*]
}
@enduml
```



This kind of usage of state diagrams should actually happen during later design stages, but it might be used like here to make it clear in what states certain human processes can be and how information inside system should describe state of users actions.

4. References

- MVP & Lean Startup
    - http://blog.ycombinator.com/minimum-viable-product-process/
    - http://theleanstartup.com/principles
- UX
    - https://en.wikipedia.org/wiki/User_experience_design
- Personas
    - https://www.smashingmagazine.com/2014/08/a-closer-look-at-personas-part-1/
    - https://creativecompanion.wordpress.com/2011/05/05/the-persona-core-poster/

- Plant UML
    - http://plantuml.com/
    - http://plantuml.com/activity-diagram-beta
    - http://plantuml.com/salt
    - http://plantuml.com/state-diagram
    - www.plantuml.com/plantuml
- GraphViz
    - http://www.graphviz.org/
- Ionic Creator
    - http://ionic.io/products/creator
- Balsamiq
    - https://balsamiq.com/products/mockups/
- BPMN
    - http://www.bpmn.org/
- PostgRest
    - http://postgrest.com/
- Postgres
    - https://www.postgresql.org/
- Sql
    - http://www.andrejkoelewijn.com/blog/2008/10/27/sql-is-a-dsl/
- JWT
    - https://jwt.io/
- Dashboard
    - https://en.wikipedia.org/wiki/Dashboard_%28business%29
    - http://blog.invisionapp.com/designing-better-dashboards/