11/11 meeting minutes

Agenda
- Discuss deliverables for the week and assign them
- Start working on deliverables

Name: ~~GroceryMates?~~ GroceryShip

Tasks assignments (due 11/15 7pm)
- Polish design (Definitions, Concepts, Security concerns) - Czarina
- Data model - Joseph, Vincent
- UI & Challenges- Chou
- Making pitch presentation, Motivation - Vincent, Joseph

https://stellar.mit.edu/S/course/6/fa16/6.170/courseMaterial/topics/topic1/resource/project-checklist/project-checklist.pdf

Design	Pitch	Highlights	Name: a short and memorable name for your app
			Purpose: why is the app needed?
			Concept: what is the key concept?
			Risks: what might go wrong?
	Overview	Motivation	Brief description of system to be built
			Key purposes (what problems does it solve? why should it exist?)
			Each purpose summarized in a short sentence and then explained
			Deficiencies of existing solutions (if relevant)
	Design essence	Concepts	Key concepts, each with: name, purpose, operational principle
			Concepts have short and memorable names
			Any anticipated misfits: problems that concept design will avoid
			Purposes explain what concepts are for
			Operational principles are scenarios showing how purpose is fulfilled
			Concepts capture central design ideas, not routine notions
			Concepts are not just actions or user interface components
		Data model	Abstract model of application state in diagrammatic form
			Explanations of any non-obvious elements
			Textual constraints included for constraints not expressible in diagram
			Insights about design explicitly listed
			Schema representation details excluded
			Syntactically valid diagram with consistent naming & layout
			Generalization used appropriately
			Names of sets and relations well chosen
			Correct use of multiplicity and immutability markings
			Avoidance of anti-patterns (eg, boolean flags instead of subsets)
		Security concerns	Summary of key security requirements and how addressed
			How standard web attacks (such as XSS, CSRF, etc) are mitigated
			Threat model: what assumptions you're making about attackers
		User interface	Wireframes for application, with explanatory notes overlaid
	Challenges	Design risks	Anticipated misfits: what might not work?
			Mitigations: how will you respond to misfits?
		Design choices	List of problems to resolve in concepts, data model or user interface
			For each problem: options available, evaluation, which chosen

Pitch

Motivation

MIT students are generally busy and don't even have the time to make it to the grocery store to buy food and other things. Some MIT students on the other hand have time to go to Shaw's, so why not use that time to make some extra money? Wouldn't it be great if we could just ask for their help and ask them to help us buy that box of chocolates we wanted as a late night snack?

Scenario

A requester makes request for an ingredient to be delivered by a certain time.

A shopper takes the request…

Shopper looks for item in the grocery...If it is not there, shopper can indicate that and release the request

If it is there, shopper indicates that items have been bought and sets up a pickup time and location

They meet up and items are delivered

Both parties indicate that the request has been complete and payment is processed within the app

Risks
● Wrong item
● Irresponsible users
● Fraud credit card (or no money in TechCash account if using TechCash)

```
            Deadline
               |
Item <----Request ---!> Requester
|              |
?              → Payment?
v
Shopper
```
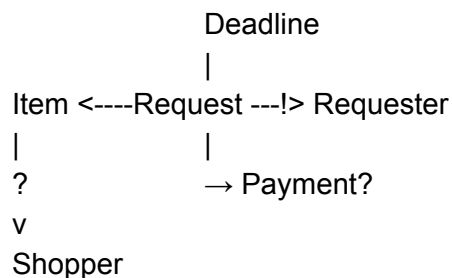
User --subset→ Requester, Shopper
→ overall rating
→ rating (0 or more ratings from other users)

Rating --for→ User
--from→ User
→ score

Item → Description
→ ???

Delivery --- pick up point
|
pick up time
→ Request
Requester, Shopper, rating

Textual Constraints:
- Requester and Shopper of a Request must be different Users.
- Delivery pickup time is before the Request Deadline.

Explanations:
…

Insights:
- Unclaimed requests will have have no shopper assigned yet, so the relation for shopper is 0 or 1 and not exactly 1.

User Interface (Wireframes for application, with explanatory notes overlaid):
App name
Registration: payment methods
Homepage**:** request and deliver (put a red circle (with a number) to indicate updates/notifications to the corresponding tab, i.e. if someone has claimed it)
Request page/tab: form for the request or your current request, status of request (has someone claimed/bought it, have they requested a pickup)
Deliver page/tab: request feed (all requests), possible filtering by user, location, etc?
Settings page
Delivery completion page: set pickup time, location
Logout, sign in, sign up buttons
Make a complaint page?

...