CENG 492 Project Presentation
by Takla Design Group
Project: Trade Center

- An android application
- Allows users to trade, sell and share items
- Similar applications out there in the market (Passage, Semt, etc.)
- Our uniqueness lies within the principle we share.
Similarities between these apps and Trade Center

• Same basic idea: A platform in which items can be sold or traded.
• Being location based (GPS information is used)
• Users can see items that are near them
• Trading and selling options for the items.
Differences, and Our Approach

- Apps out there, all have static approaches
- Not enough functionalities, or interactions with users
- This kind of approach pushes user out of the picture
- Our biggest difference; a dynamic approach
- An easier-to-use app with more functionality
- User at the center
Dynamic Functionalities

- Interests / needs, determined by tags
- Advanced search option with more parameters
  - Tags, trading options, location etc.
- Keep them notified: A notification
- Option based on the user's interests / needs.
- Recommendations based on users' interests and location information
Dynamic Functionalities - Cont’d

• Addresses: Users can store and specify addresses upon an item upload (item has the address specified by user)

• A UI demonstration with some tabs from the application:
Tools Used in the Project

Amazon Web Services

Node.js

Android Studio

MongoDB
Project Structure

- Architecture Diagram
Project Structure

- RESTful architecture is being used
- Client and server sides are separate and does not know about each other
- Communication between front-end and back-end is provided by HTTP requests/responses
- JSON is preferred as the markup language
Front-End Development

- Android Studio as IDE
- Java as programming language
- Windows as OS
- Google Cloud Messaging (GCM) for Instant Messaging and Google Maps API for modules which require user's location
- Gradle is being used for build system
Front-End Development

- Application mostly advances by state transitions
- About 30-35 classes
- Fragments are being used for Messenger module, rest is mostly Intents
- User presses buttons or other clickable widgets in order to switch between these Intents (or Fragments)
- ListView structure for displaying images, Navigation Drawer for User Profile
Back-End Development

- Amazon EC2 Web Hosting as Infrastructure
- Node.js for handling requests and interacting with DB
- MongoDB for database management
- Mainly used NPMs (Node Package Modules): Mongoose, ExpressJS, Forever, Connect, BodyParser, Crypto (and etc.)
- Linux as OS
Back-end Development

- One executable file --> app.js
- Major module dependencies, configuration and routing requirements done in app.js
- One router file --> routes.js
- It is for handling requests/responses and their related datum with respective functions
- About 20 functions at the bottom layer (login, register, item upload, send message etc.)
- They interact with DB, send responses to routes.js
Conclusion

• A shared idea, but different approaches
• Aiming for user convenience
• Interaction and dynamicity is the key
• User in the center
• A simpler and more compact UI
• That's what Trade Center is all about!!!