Project Plan

Browser Sharing for Customer Support

The Capstone Experience

Team Amazon

Liyuan Duan
Megha Erappa
Jonathan Kushion
Rahul Yalamanchili
Eunice Yoon
Colin Zhong

Department of Computer Science and Engineering
Michigan State University
Spring 2019
Functional Specifications

• Co-Browsing between customer and support representative with no installation required.
• The customer joins a queue for help with the press of a button.
• The representative can see exactly what the customer sees, and can chat to work through the problem.
Design Specifications

• Recorder
  ▪ Capture DOM events that are happening.
  ▪ Remove private customer credentials.

• Mediator
  ▪ Send and receive customer interactions with the webpage.
  ▪ Keep messages recorded from representative and customer.

• Replay
  ▪ Display events onto representative’s side.
Screen Mockup: Customer Assistance Button
Screen Mockup: Co-Browsing Notification

Warning!

By clicking confirm you agree to co-browse with a qualified representative at Amazon.

They will be able see your screen and chat, but will you will stay in control of your computer.
Screen Mockup: Support Queue

Tom Doe

Time in queue: 5:26

Queue for co-browsing:
- Tom
- Jerry
- Larry
- Terry
- Barry
- Carrie
- Shawn

Co-Browse
Screen Mockup: Chat (Customer)
Screen Mockup: Chat (Support)
Technical Specifications

• Amazon Web Services (AWS) EC2 Server
  ▪ Used to store session data into the AWS database.

• AWS RDS: SQL Database
  ▪ Contains the co-browsing sessions’ data.

• Fiddler - Web Debugging Tool
  ▪ Used to inject our recording script into websites.

• Swagger - API Tool
  ▪ Used to create the REST API for receiving and sending DOM event data.

• JavaScript
  ▪ Record and replay of DOM events within browsers.
System Architecture

- Customer
  - Record
    - JS
  - EC2
    - Load Balancer
    - Recorder
    - Mediator
    - Database (AWS RDS)
    - Replay
  - EC2
    - Load Balancer
    - Replay

- Support
  - Replay
    - JS
System Components

• Software Platforms / Technologies
  ▪ AWS Elastic Compute Server
  ▪ AWS Elastic Load Balancer
  ▪ AWS API Gateway
  ▪ AWS RDS: SQL Database
  ▪ JavaScript
  ▪ Browser APIs (e.g. W3C DOM API)
Risks

• Fidelity of capturing DOM events
  ▪ All events need to be captured for playback accurately.
  ▪ Research and find out what, if any, events need special cases.

• Cross browser support
  ▪ DOM events could be different for different browsers.
  ▪ Translate DOM events into an object that is universal.

• Amount of data to replay
  ▪ We have to capture and send the web page for display.
  ▪ Sending a snapshot of the web page at first and only the changes that occur after.

• Fidelity of DOM replay
  ▪ We must replay events exactly as they happened for the user.
  ▪ Research the accuracy of replay and make positions relative.
Questions?