University of Colorado Boulder

DIGITAL WEAVING

Project 6 - Report

ECEN 5783- EMBEDDED INTERFACE DESIGN

FALL 2019

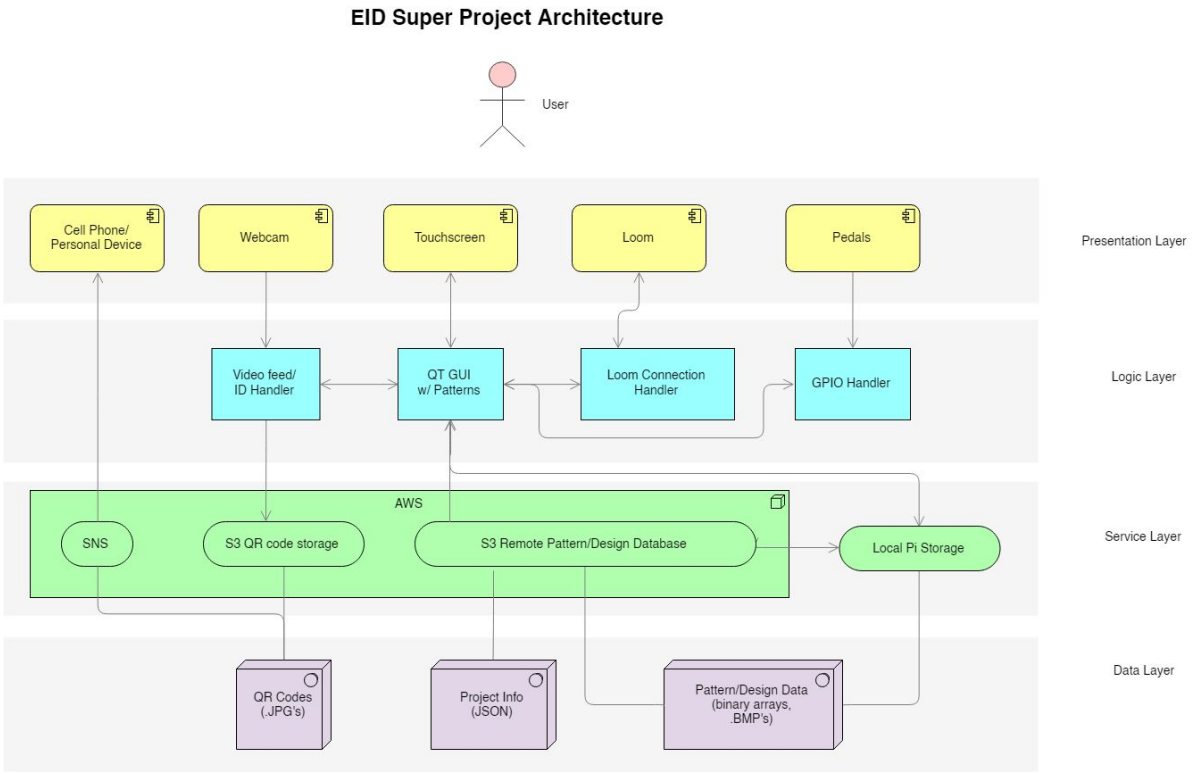
Custom Super Project: Rapid Prototyping for Smart Textiles UI

Smitha Bhaskar and Shanel Wu

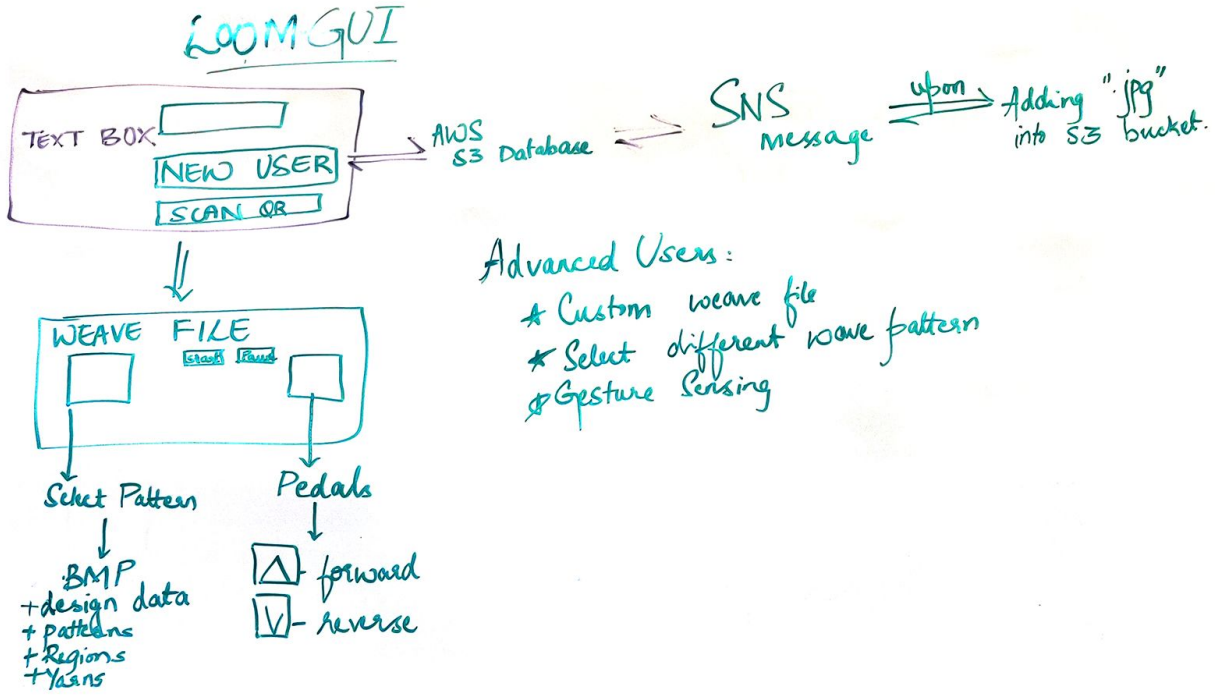
Github Link:

https://github.com/SMITHABHASKAR/EID_ECEN5783_Fall19/tree/master/SuperProject

Final System/Architecture Diagram:



GUI sequence diagram:



Project Deviation

- **Change:** AWS integration (we did more locally on Pi)
    - QR detection
    - Only pushing weave design data at beginning/end of work sessions

We have a file called aws.py which uploads the qr code to the bucket on the AWS S3. S3 is an AWS Storage service which has a lot of features which can be customized as per the project. This is a snippet of the code. The file does not contain the Access Key and Secret Key which link the code to the S3 bucket created in my account.

Code Snippet:

```
def upload_to_aws(local_file, bucket, s3_file):
    s3 = boto3.client('s3', aws_access_key_id=ACCESS_KEY,
                      aws_secret_access_key=SECRET_KEY)
```

- **Added:** Experimented with camera swipe gesture sensing for editing weave design
- **Missing:** annotation features, template selection, touchscreen

Third-Party Code

- Python libraries
    - OpenCV/PIL
    - Numpy
    - QT and PyQT
    - Qimage2ndarray
    - PyPi
    - PyZbar
- Collaborators
    - Lea Albaugh - TCP connection, originally written in Processing
    - Mikhaila Friske - patterns JSON from AdaCAD

Project Observations

- ***Getting and Using User Feedback:*** We worked with our users in the Unstable Design Lab throughout the design and prototyping process. Without this continuous feedback, our prototype would not have been so successful. During the project proposal, we used a brainstorming session during a lab meeting to determine the weaving UI's core interactions (e.g. using pedals to progress through a project, and wanting an on-loom display like a touchscreen), and the current UI's most egregious problems. When conducting a paper prototype test, we received feedback from both new and experienced users that gave us insight into developing a beginner-friendly, streamlined interface that was still flexible enough to excite advanced users. While building the final prototype, we presented updates to users to make sure that the UI was aligning with their expectations.
- ***Hardware/Platform Problems:*** We encountered some setbacks because of how fragile the Raspberry Pi is. We broke one Pi by accidentally shorting its ground and power pins. We almost broke its replacement when we left it sitting on a carpet floor; apparently, a spark of static electricity can do bad things to a Pi. Fortunately, the second one was recovered by reformatting the SD card. Knowing the Pi's sensitivity, the first step of any future project will be mounting the Pi on a protective base. Also, installing OpenCV on the Raspberry Pi was extremely time-consuming and complicated. The PyPI package from "pip install" does not properly install on Raspbian, so we had to manually compile the binaries. Furthermore, OpenCV seems to interfere with the Pi's video drivers for the touchscreen.

- ***Designing DIY Pedals:*** We obtained stomp switches for the prototype pedals, the same kind used by musicians and circuit benders to create custom guitar effects. The heavy duty switches provided great tactile feedback for the users, even before we made the pedal housings. The housings for the pedals were designed in Adobe Illustrator, consisting of a base and a top plate. The top plate adds surface area to the switch button, making it more comfortable to press with a foot. The base tilts the top plate at an angle, letting the user easily press the pedal with their toe and making any icon on top of the pedal more visible.
- ***Structuring the GUI Code:*** Early on in our prototype development, we realized that the GUI screen that the user sees when weaving would host most of the core functions, such as communicating with the loom and pedals. As the code associated with that screen got longer and longer, we realized that we had to break up the file into sections. There were still several functions to implement, which would only add more code. We separated the main QT window code from the custom widgets that we created to display weaving patterns. The loom and pedals each needed their own file to handle their connection protocols. Throughout our prototyping process, we tried to keep our code compartmentalized in this way to make debugging and future expansion easier.

****