Money Mobile: Kiva App Design Proposal

Matthew Pool

Computer Science Undergraduate

Southern New Hampshire University

CS 319: UI/UX Design & Development

Professor Peter McNally

February 25, 2024

Kiva and *Money Mobile*

Kiva is a 501 non-profit organization that allows low-income entrepreneurs, students, and others living in underserved communities to work with lenders across nearly 100 countries (Kiva, n.d.). The *Money Mobile* app is designed to assist borrowers and lenders in making this process easier and more intuitive, using industry standard UX (user experience) guidelines like those created by Google and Apple. Development of a mobile app for borrowers, in addition to recommendations for a cloud-based variant for lenders, will be discussed in further detail.

Borrower App

Borrowers

The mobile app, *Money Mobile*, allows borrowers (and potential borrowers) to apply for and track loans financed through *Kiva*. The majority of loans – over 4000 of them at the time of this article – are given to citizens of Africa and Asia, primarily for agricultural and food-related costs (according to current data at www.kiva.org) Because of the current needs and available infrastructure in impoverished communities in parts of Asia and Africa (Eurasia Review, n.d.), this app has been designed specifically for users expected to access the app using technologically-limited embedded devices (i.e. flip-phones). Other versions will use right-to-left (RTL) directionality for UI elements to accommodate cultures in parts

of the Middle East, Asia, Africa, and other regions; however, numerical values and the *Kiva* and *Money Mobile* logos will maintain their existing orientations (Material Design, n.d.). The app will be localized, based on locale preferences, to appropriately customize directionality and currency symbols. Please take a moment to review the attached wireframe diagram, if desired, to familiarize yourself with the application's UI before details are discussed in the following text.

User Interface (UI)

The *Money Mobile* app will be created using a responsive and reactive layout, so whether using portrait or landscape orientation or a smaller screen (like those on traditional flip-phones), UI elements will scale and reposition accordingly. The app will also include the *Kiva* logo on the left end of the status bar, located at the top of each screen. The logo acts as a hyperlink to www.kiva.org and gives a satisfying "click" when pressed. "Money Mobile" text appears centered on the status bar, using a stylized font specific to the *Mobile Money* brand. All other text will use a normal Sans/Serif font to provide consistency with Android OS (Material Design, n.d.).

In the bottom-right corner of each screen is a menu icon, consisting of three vertical dots as recommended by Google (n.d.), appears that allow a pop-up dialog to display with an account icon and text link. The icon and text links each navigate to the *Kiva* account screen. Under these links is a light/dark mode toggle switch, which defaults to the current system settings and provides a high-pitch and low-pitch "click" when turning dark

mode on and off, respectively. The color scheme will match *Kiva*'s dark/light green branding colors, as allowed by the *Kiva* Code of Conduct (Kiva, n.d.). Turning on dark mode interchanges the primary color and accent color and inverts any black/gray/white text and background colors. Finally, a "Back" link and "Donate" link appear at the bottom of this menu dialog. The "Donate" link navigates the user to the *Kiva* donation screen to help support *Kiva*'s crowdsourcing model. The "Back" button simply closes the dialog and navigates back to the previous screen.

Login Screen

The login screen consists of "Login" in large, stylized font, followed by "to access your *Kiva* account" underneath in smaller text. Down farther, a "Username" label and text entry box appear above a "Forgot Username" link. This is followed by a "Password" label and text entry box, which obscures the input text, as well as a "Forgot Password" link. Finally, a "Submit" button to submit the login credentials to *Kiva*'s database servers via the *Kiva* API is shown to authenticate the user for account authorization purposes. A *Kiva* account will need to be created if none exists, with the "Sign Up" link, located directly under the "Submit" button, providing this process. Upon triggering the link, the "Login" title will be changed to "Sign Up", and the "Sign Up" link will be changed to "Log In". The bottom text is replaced with a message that includes a Kiva text hyperlink to www.kiva.org and says, "Money Mobile helps those in need borrow through Kiva – a non-profit."

Pressing any of the "primary" (green) buttons at any time emits a deep, subtle "click" sound for extra dopamine release!

Main Screen

Following a successful account login, the user is presented with the main screen of *Money Mobile*, along with a quick, melodically-friendly "wind-chime" sound, with "Loans" rendered in the same stylized font as the title text on the previous screen. Under the "Loans" text, in smaller, bold text, three columns identify the most crucial data for each loan:

- Loan ID
- Amount Due
- Due Date

The loans are ordered chronologically by due date. If no loan exists, the message "Select the 'Borrow' button to get started!" appears instead. The "Borrow" button is centered near the bottom of the screen and is the most prominent button on this screen. Underneath each amount due value is a "PAY NOW" button, followed by a "Details" link to the very right of each. Although the most sought-after information is found on the main page of the app for convenience and usability efficiency, further details regarding the loan, lenders, and related information can be found by selecting the "Details" link. If multiple loans exist, the list of loans allows for vertical-scrolling, if needed. At the far-right end of

the status bar, a notification bell is displayed that implements a blue notification dot (along with red text) to indicate recent updates and messages. Pressing the bell triggers another subtle, but satisfying, "click" and opens the notification list, with notifications reacting to typical touch/swipe gestures.

Data Integrity and Security

The loan-specific details, like the account details, are accessed via the *Kiva* API (application programming interface), which accesses *Kiva*'s database servers, and is used to keep all account/loan information up-to-date. Data can be cached locally on the user's device, whenever the user opens the app, makes a payment, requests a loan, or any information is updated on the server. Local files will be encrypted using AES (advanced encryption standard) for securing stored data and data in transit, along with requests using HTTPS/TLS (hypertext transfer protocol secure/transport layer security). SHA (secure hash algorithms) will be used for login credentials transferred between the user interface and *Kiva* API. Optionally, a CDN (content delivery network) can be implemented in various locations to allow caching for lower bandwidth infrastructures, as bandwidths of only 50 Kbps to 384 Kbps (2G or 3G, respectively) may exist as a limiting factor.

Accessibility

Black text will have ~87% opacity for high-emphasis, ~60% for medium-emphasis, and ~38% for low-emphasis, when shown against a white background. Colored text and

backgrounds will consist of at least a 3:1 contrast ratio (for large text) and a 4.5:1 contrast ratio (for normal text), as recommended by the WCAG (Web Content Accessibility Guidelines), which is created and maintained by the WAI (Web Accessibility Initiative) of the W3C (World Wide Web Consortium) (Level Access, n.d.). Color tones will not be finalized until contrast ratio is verified by a tool like WebAIM Contrast Checker (WebAIM). Alternative text will be provided for input fields, buttons, and other essential information to assist users who rely on screen readers. In bold, red caps, directly under the notification bell, a clickable label with the text "MAIL" is replaced with the text "NEW" (along with the blue notification dot) when a new notification is received. The use of textual and color redundancy in the notification system provides greater inclusivity for those with various color blindness and other visual impairments. This is achieved by not relying solely on color alone to convey information (We Are Colorblind, n.d.) and keeping icon labels visible at all times for clarity (Harley, A., 2014).

User Experience (UX) Design

This UI was designed with the user (borrower) needs and requirements in mind. I believe most people will either be applying for a loan or checking an existing loan to find the balance due and due date and optionally make a payment. This is the information I have included on the main screen, attempting to minimize the number of UI elements and steps to perform simple tasks, to account for the hardware and infrastructure limitations imposed by these embedded devices and create an effective and efficient UX. The loan ID

for each loan is also displayed, so that the user may distinguish between multiple loans (if present). The message to "Select 'Borrow' to get started" is shown to guide the user, in the case no current loans exist for the user. The "BORROW" button is centered near the bottom of the screen and is sufficiently sized for easy access and discovery. The "Borrow" button, "PAY NOW" button, and "Details" link are spaced apart enough to make the process less prone to user error.

Another user need, as shown in the borrower's user story, is to have access to information about *Kiva* and borrowing. The *Kiva* logo (in the status bar) serves as a hyperlink to *Kiva*'s website, to satisfy this need. The notification bell provides a way to notify the user of any important updates or new information, so that they can stay informed, as requested. The details screen (accessed via the "Details" link) provides all the loan information required to track the loan payback process, as needed by the borrower persona.

System Architecture

The underlying framework will be constructed using *Android Studio* and the *Java* programming language, in conjunction with the *Retrofit* library for asynchronous/synchronous network requests and RESTful web services running on background threads, to allow for a smooth and performant experience. The MVVC (Model-View-ViewController) architecture will be used in conjunction with a Repository pattern database access, with priority given to any limitations and restrictions imposed by the *Kiva*

API and Code of Conduct. This tiered architecture will follow Google's modern app recommendations, which advise using a layered architecture and unidirectional data flow (UDF).

Lender Cloud Application

System Design

Using these same UX design principles and guidelines, a more sophisticated app can be created for *Kiva* lenders, based on a cloud architecture instead. However, even though cloud-based systems have the capacity to scale and be more sophisticated than limited embedded systems, *Kiva* is a nonprofit, and scaling data in the cloud can be expensive. Therefore, it is recommended that much of the system UI code and input validation be integrated into the client's device, to cut down on cloud costs; however, personally sensitive account information (passwords) should remain on encrypted servers instead.

Various tabs can be used to show lenders the distinct categories of information regarding borrowers, loans, and lending, in order to help facilitate the decision-making and transactional processes that they require. Because the technological components involved in cloud-based computing will be more powerful than its flip-phone counterparts, animations can be implemented into the lender application for tasks like opening the notification list or menu, to provide a greater user experience. More complex colors and

other UI techniques can be integrated into the UI elements, such as shadows to help show hierarchy and distinctness among UI components.

System Architecture

Using a cloud-based system will allow for efficient and affordable scalability, as the primary data storage and processing can occur on powerful remote servers in the cloud. The client application will access these resources via a standard internet connection – an important constraint to consider, as it can add a small amount of latency to the user flow. To circumvent this potential issue, UI elements with nonsensitive information can be kept locally on the user's device. The cloud platform will also allow for universal updating of system software, as it can be done automatically server-side. By implementing these various design and development strategies, a more secure and robust software product can be created and provide a joyful and helpful user experience to all users!

References:

Eurasia Review. (n.d.). *News & Analysis*. https://www.eurasiareview.com/.

Google. (n.d.). *Android Developer*. https://developer.android.com/design.

Harley, Aurora. (July 27, 2014). *Icon Usability*. Nielsen Normal Group. https://www.nngroup.com/articles/icon-usability/.

Kiva. (n.d.). *Code of Conduct*. https://www.kiva.org.

Level Access. (n.d.) *WCAG (Web Content Accessibility Guidelines)*. https://www.levelaccess.com/compliance-overview/wcag-web-content-accessibility-guidelines/.

Material Design. (n.d.). *Material Design.* https://material.io/.

We Are Colorblind. (n.d.). *Designing with the Colorblind in Mind will improve your Design for Everyone*. https://wearecolorblind.com/articles/designing-with-the-colorblind-in-mind-will-improve-your-design-for-everyone/.

WebAIM: Web Accessibility in Mind. (n.d.). *Contrast Checker*. https://webaim.org/resources/contrastchecker/?fcolor=000000&bcolor=EEEEEE.