GitBegin:

A simple and easy-to-use interface for beginning in git

**Windows Team:**

James Smith

Michael Kelly

**Android Team:**

Alexander Canton

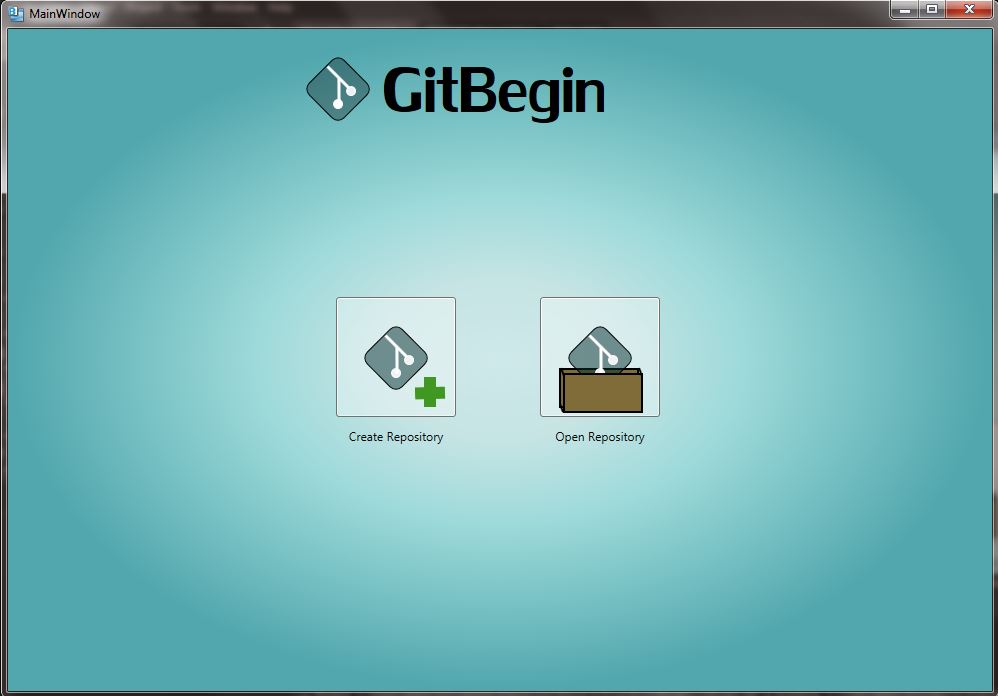
Bibek Ghimire

**Windows Design Decisions:**

The windows side of our application was designed using Microsoft’s Expression Blend software, which uses the WPF platform with C#. We chose this framework because it is a quick and easy way to prototype and implement an interface.

Our aesthetic design for this interface was based on a cool blue color, using our variation on the git logo. We wanted to keep it as simple as possible, as that is the main purpose of our application.

**Main Menu:** James



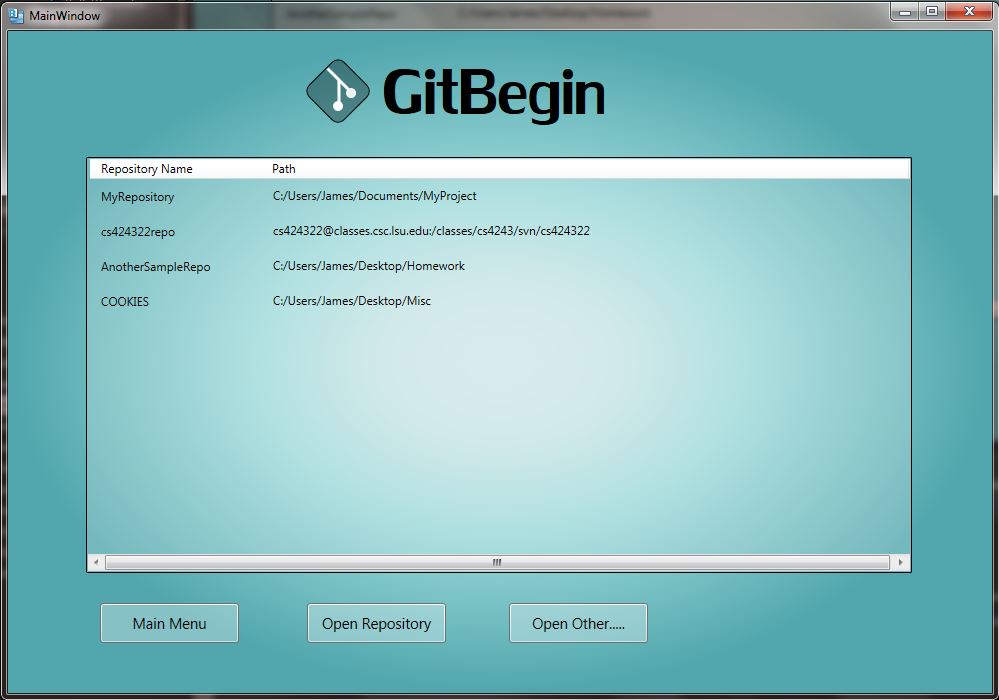
This is our main menu screen, where you can either open an existing repository or create a new one. This is by far the simplest screen in the windows application.

**Repository Creation Screen:** James



This is the repository creation screen. Here, the application lets you choose the name and full path of the repository. You may also choose to initialize it as a bare repository.

**Select Repository** James



This screen is displayed when you try to open a repository. It shows you the path and name of the repositories that you have opened. You can also add a repository that is not yet in GitBegin using the “Open Other....” button.

**Open Other:** James



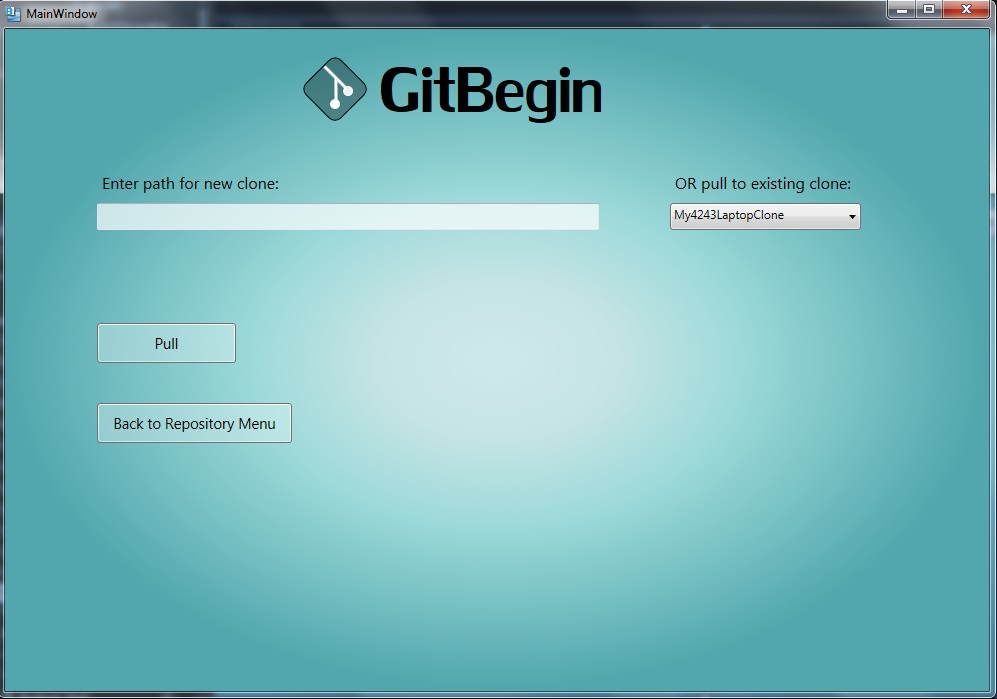
This screen lets you open an existing repository not yet recognized by GitBegin. Type in the repository’s path and name to open it.

**Repository Main View:**  Michael



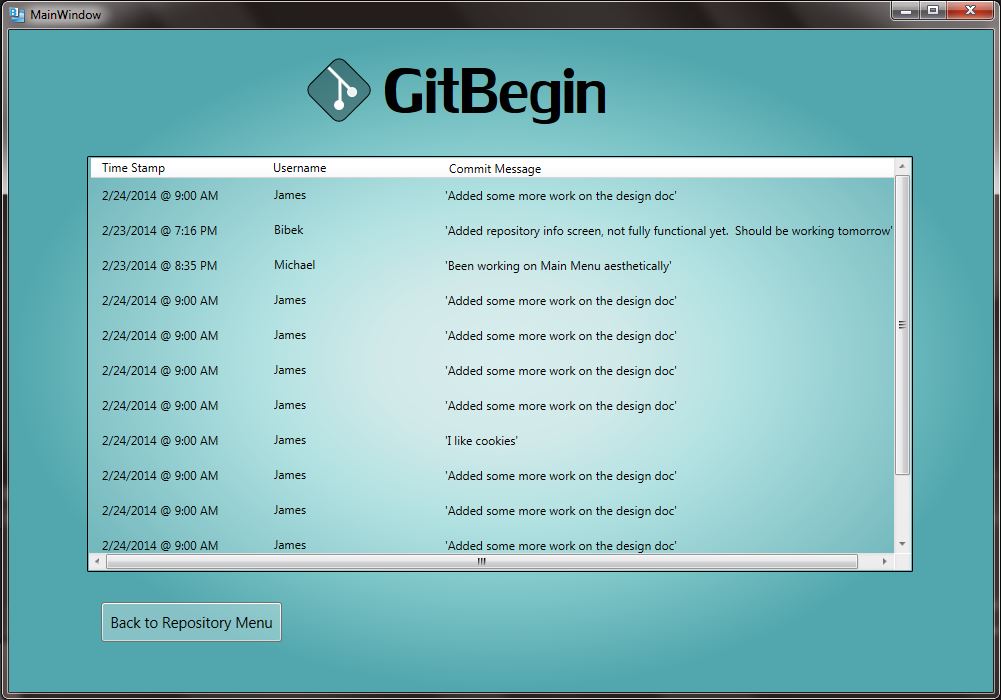
This screen appears once a repository has been opened. The user can select a clone of the repository to work with and pull, push, or commit changes from this screen or view the commit history.

**Pull:**  Michael



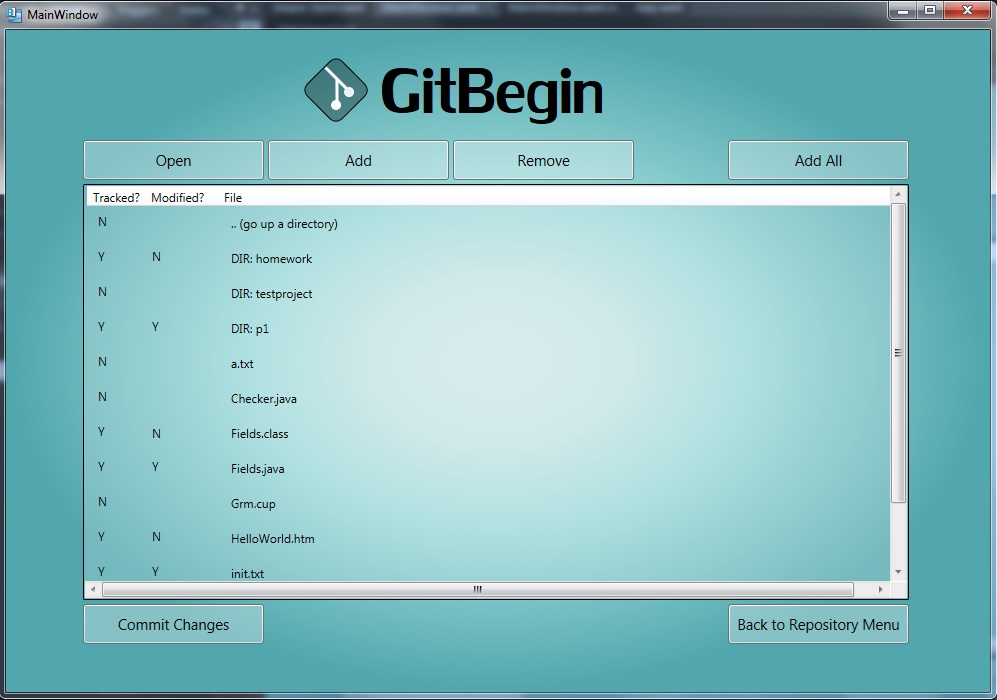
From this screen the user can either create a new clone of the active repository in a specified directory, or pull changes from the main repository to a specific clone.

**Repository History** James



This is the repository history screen. Here you can view all commits pushed to the repository using their respective time stamp, commit message, and username that committed it.

**File Viewer Screen:**  Michael



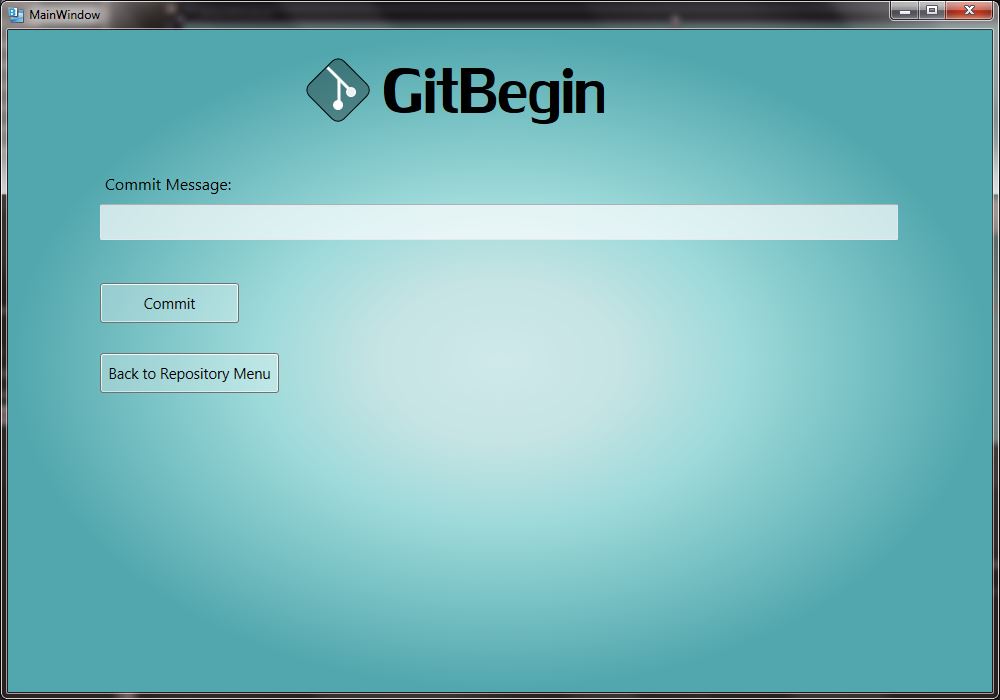
From this screen the user can browse the contents of a repository and check the status of specific files. The user can also add files to be tracked by the repository, stop tracking them, or go to the commit screen.

**Login Screen** James



This is the login screen for the application, as you must login using your git username in order to open a repository.

**Commit Screen** James

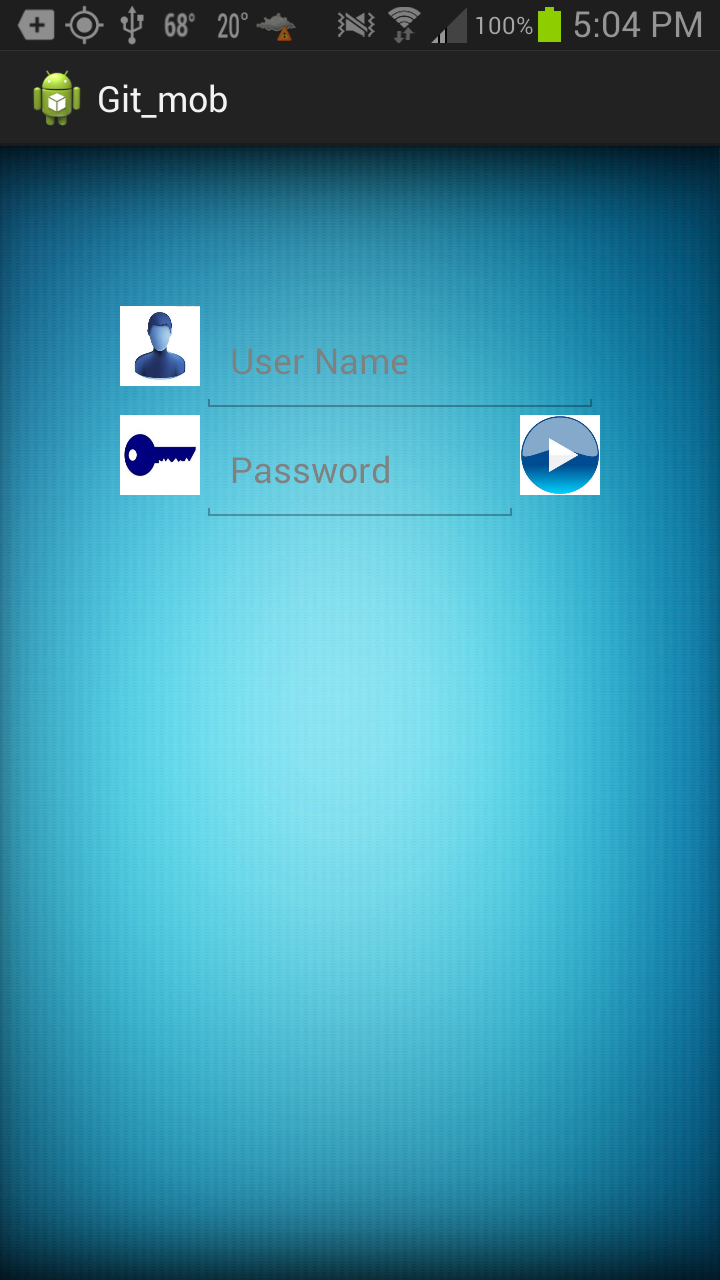


This screen appears when you choose to commit to your repository. On this screen, you set a commit message and then commit. Simple.

**Android Design Decision:**

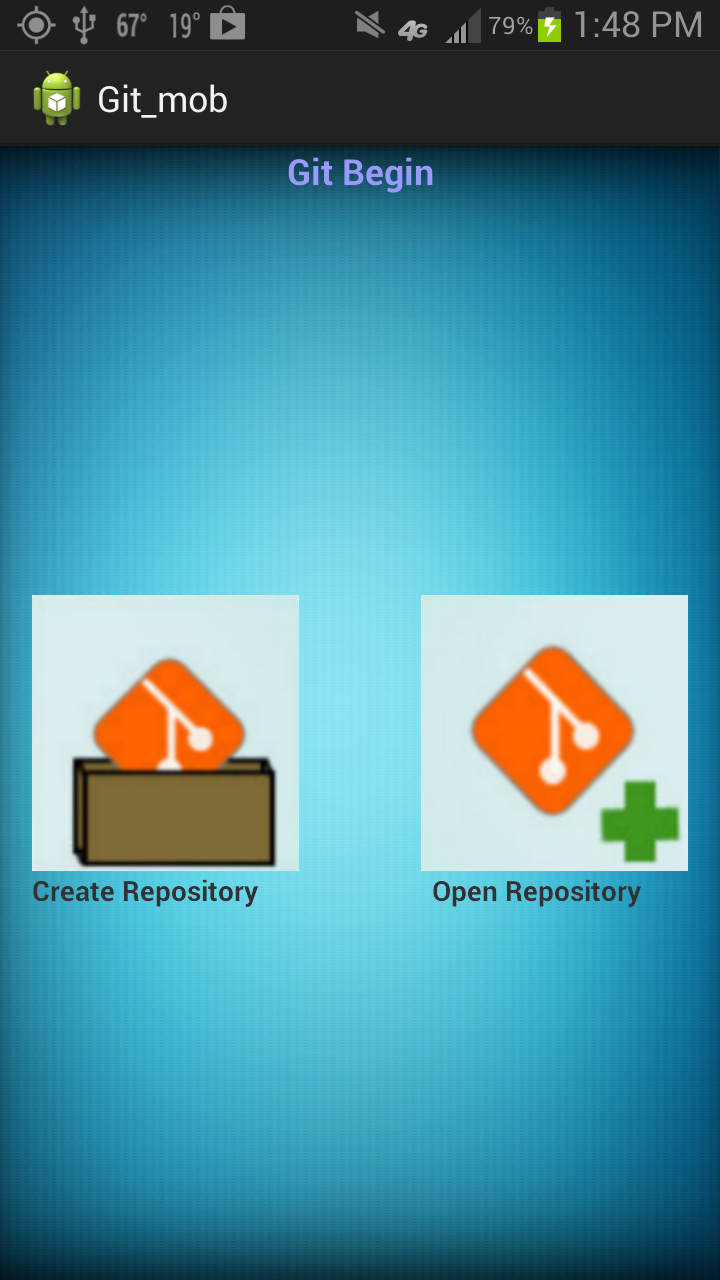
We chose android for developing application for the mobile device because its open source nature. The layout of the application are made from xml and the logic part of the application will be done using java. The java language was chosen for ease of use. Java is the native language used for the Eclipse IDE and android development. Additionally, the java language is very useful for cross-platform development. A cool blue color was chosen for the app theme. The first screen of the app will be the login page.

**Login Page:-** Bibek

This is the login screen of the GitBegin Android version. This screen leads to home screen. User is allowed to enter username and password for the access to the next screen. 

This next picture is the home screen for the app. From this place we go either create new repository or open an existing repository, pressing the two buttons

**Home Screen:-** Bibek



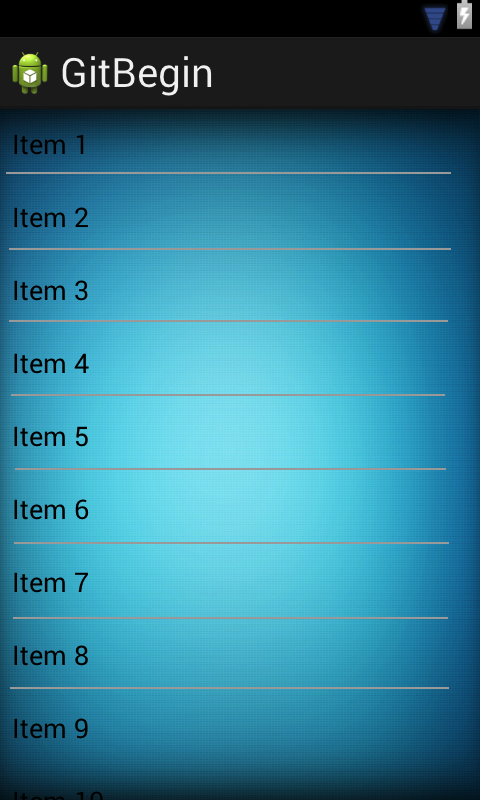
It is the home screen for the application. We reach here after the login pass. Here we can either create a new repository or open existing repository. If someone click a new repository a new view will come there and type name of repository you want to create and if someone clicked on open existing repository we are presented with a another view that contains info about the repository.

**Create Repository Screen:-** Alex



The create repository is the screen that transitions after the create button is pressed on the home screen. In this view the user can create and initialize a git repository on their local machine and link it to a remote. There will be two interactive text fields and a button. The button will have the same functionality as the enter button on the keyboard.

**History Screen:-** Alex



This view displayed the history/news screen. This is where all events within the working copy are displayed. The items will be encapsulated in a listview where that will display which user created the event, what the event was, and to which branch it was done. Each of those items will contain a hyperlink that will point the user to individual screen for the user, action, or branch. When an item is selected the screen will transition to that specific event and will display the event as well as any commit messages that may have been entered.

Example: acanton *pushed* to *Repo/development* - 2hrs ago