# Technical Specification / Schedule Screen Capture Multi-image Output Accessory

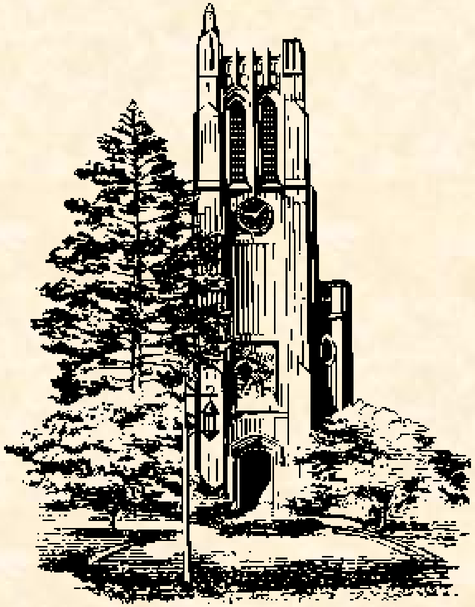
Team 11: TechSmith

CSE 498, Collaborative Design

Flavio Cheng
Greg Heil
Randy Schott
Michael Shortt
Chris White

Department of Computer Science and Engineering
Michigan State University

Spring 2007



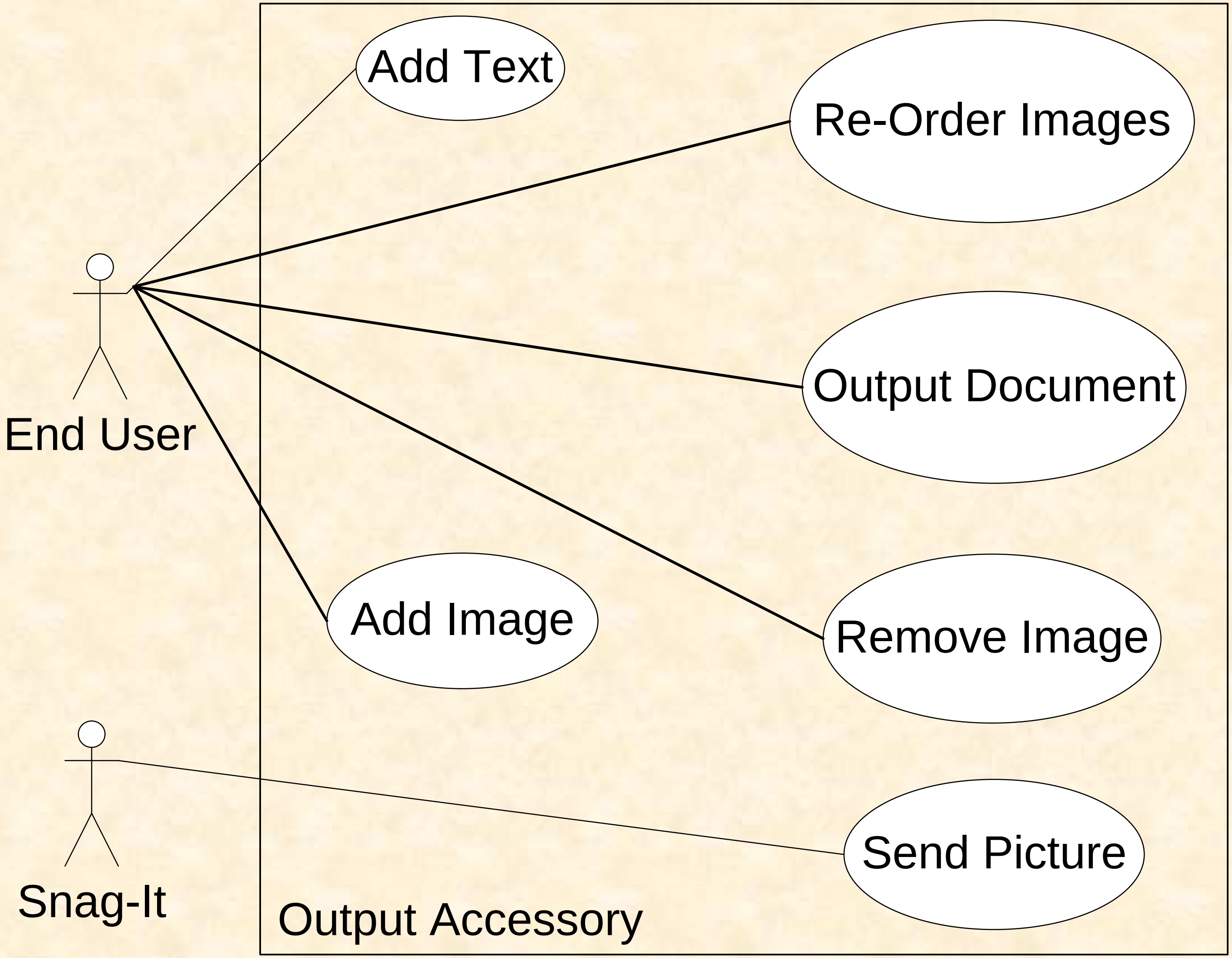
# Project Overview

- Create an “output accessory” for SnagIt screen capture application.
- Will plug-in to SnagIt’s existing output plug-in architecture.
- Allow users to collect a number of screen captures, optionally add text around the images, and output the sequence of images into a page based file format.
- Support formats HTML, Adobe PDF, Microsoft Word 2007, and WPF.

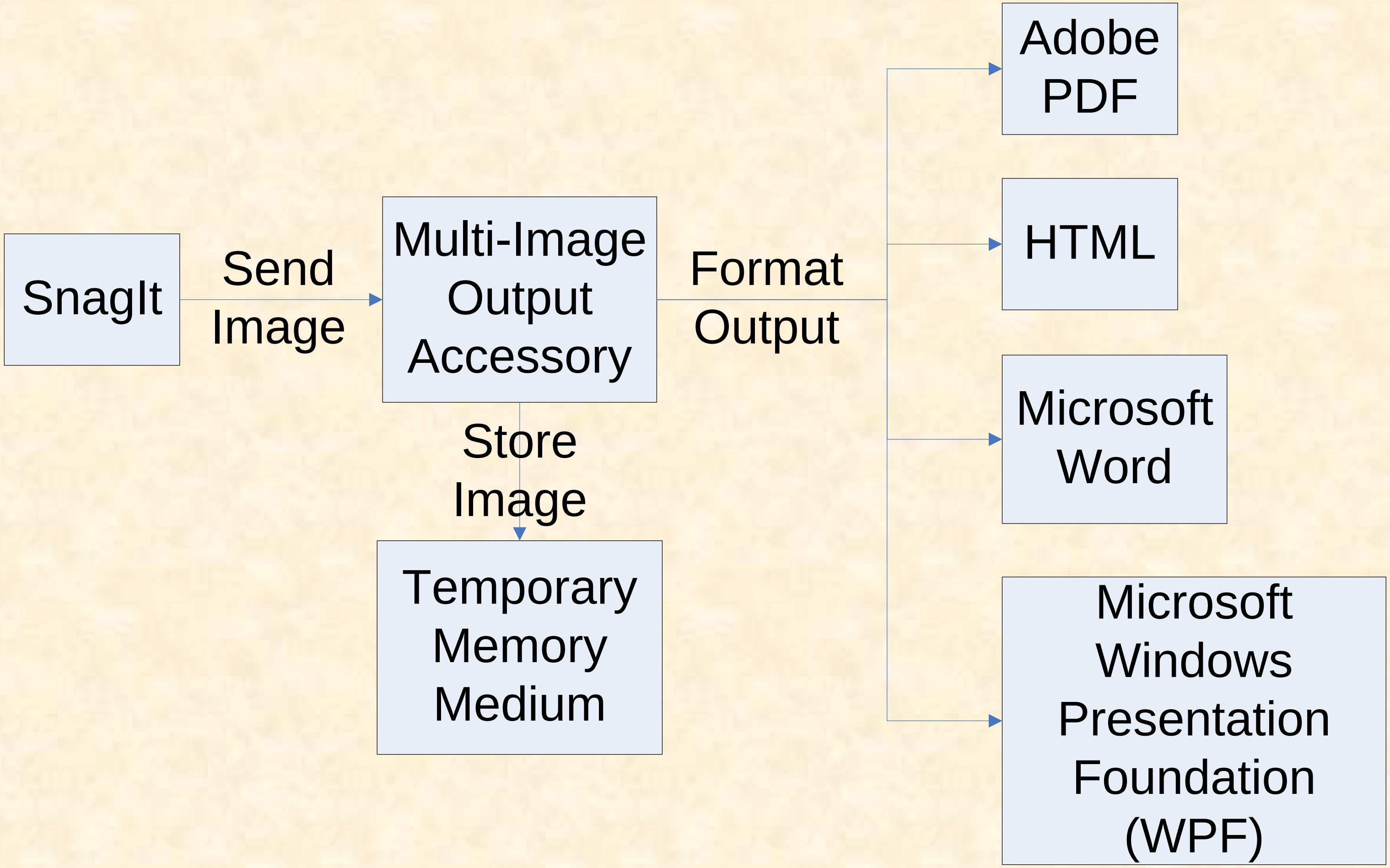
# Architecture Components

- Hardware Platforms
  - Should run on any machine that SnagIt runs on.
  - 400 MHz processor, 64 MB RAM.
- Software Platforms / Technologies
  - Requires Internet Explorer 7 with .NET 3.0 on Windows Vista/XPSP2 to view WPF format.
  - Microsoft Word 2007
  - Uses COM interfaces to communicate with SnagIt
  - XML will be used, specifically for outputting the different formats of documents.

# Architecture Illustrated
# Use Cases



# Architecture Illustrated
# Data Flow



SnagIt

Send Image

Multi-Image Output Accessory

Store Image

Temporary Memory Medium

Format Output

Adobe PDF

HTML

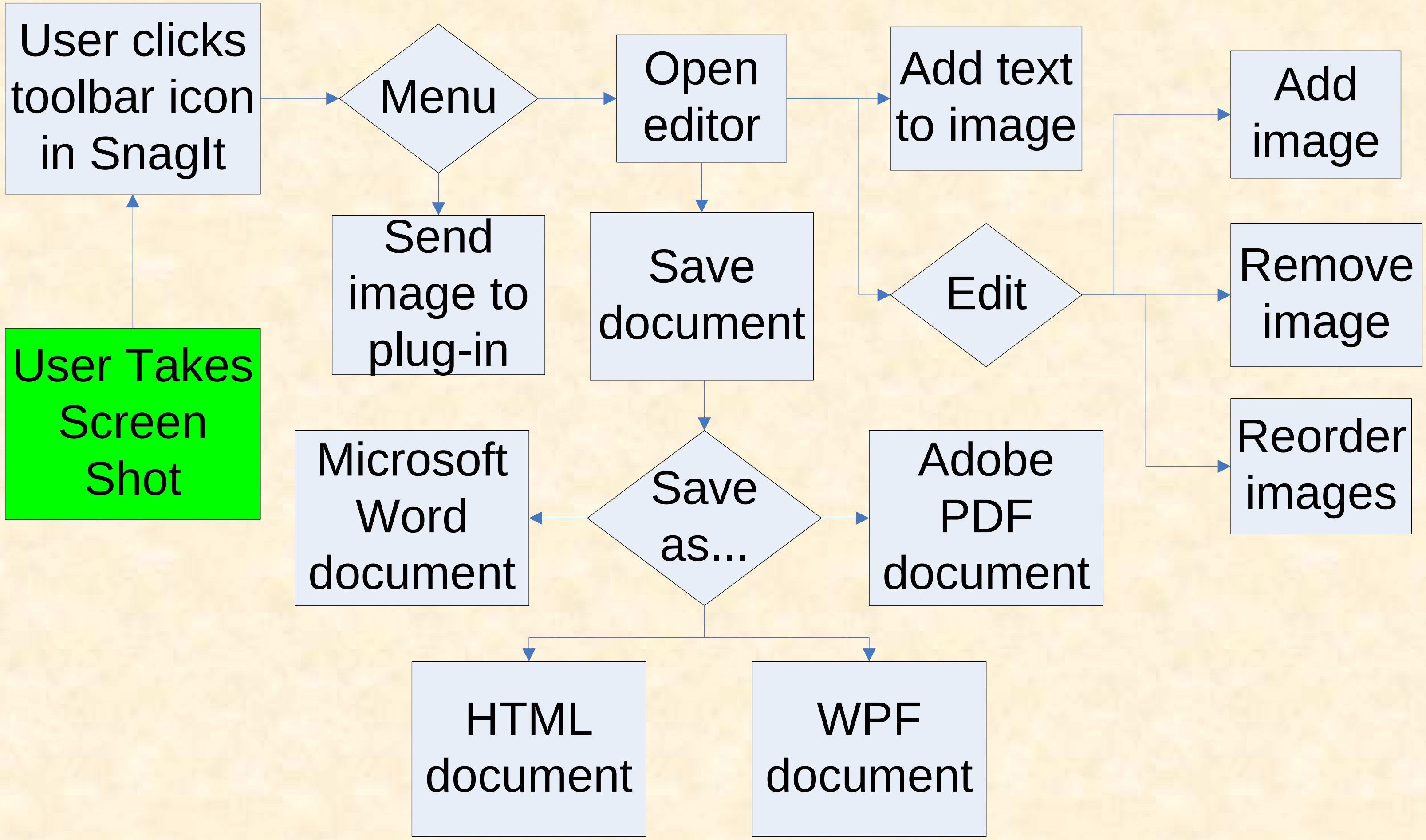
Microsoft Word

Microsoft Windows Presentation Foundation (WPF)

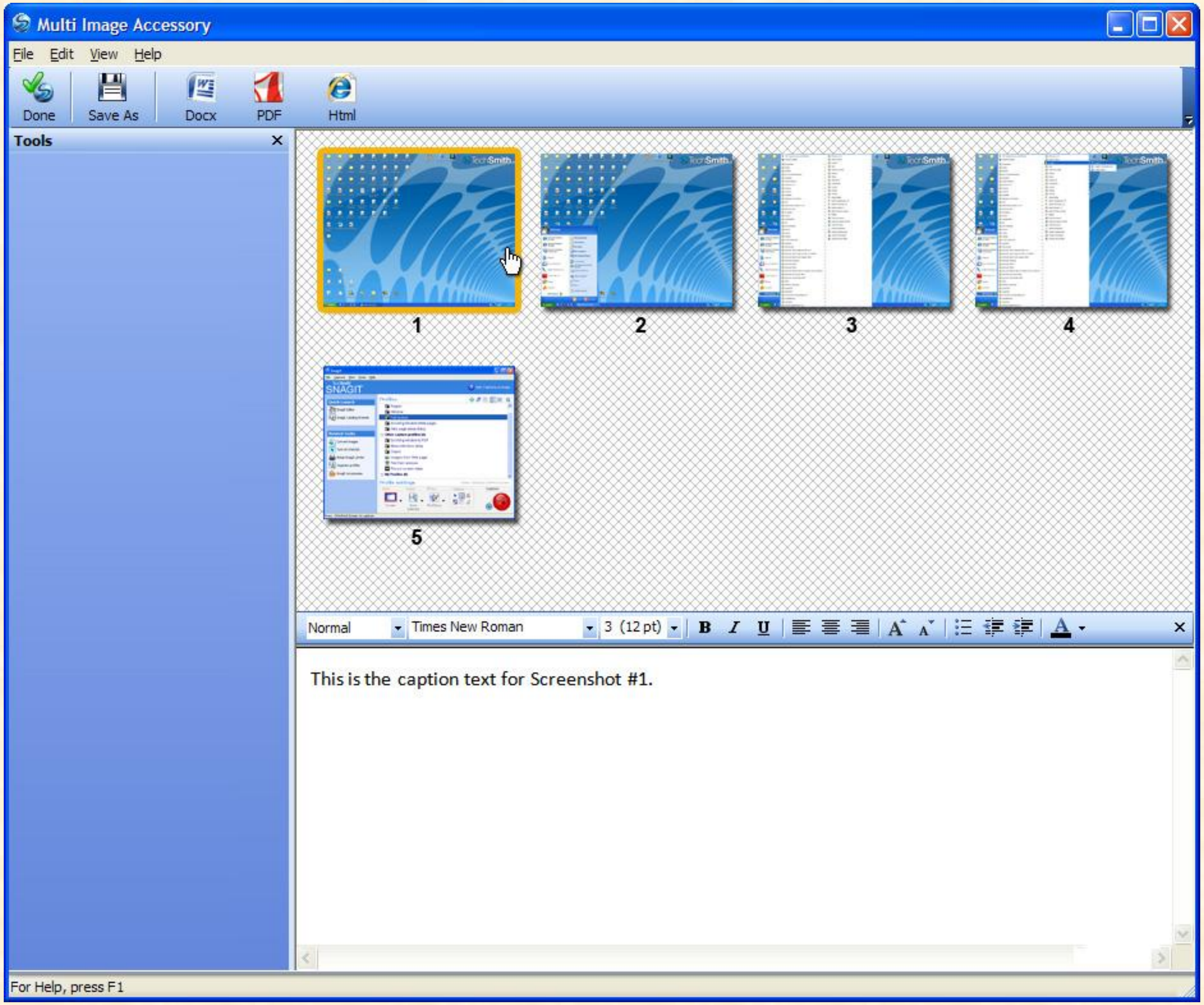
Team 11: TechSmith

# Architecture Illustrated
# Process Flow

Team 11: TechSmith

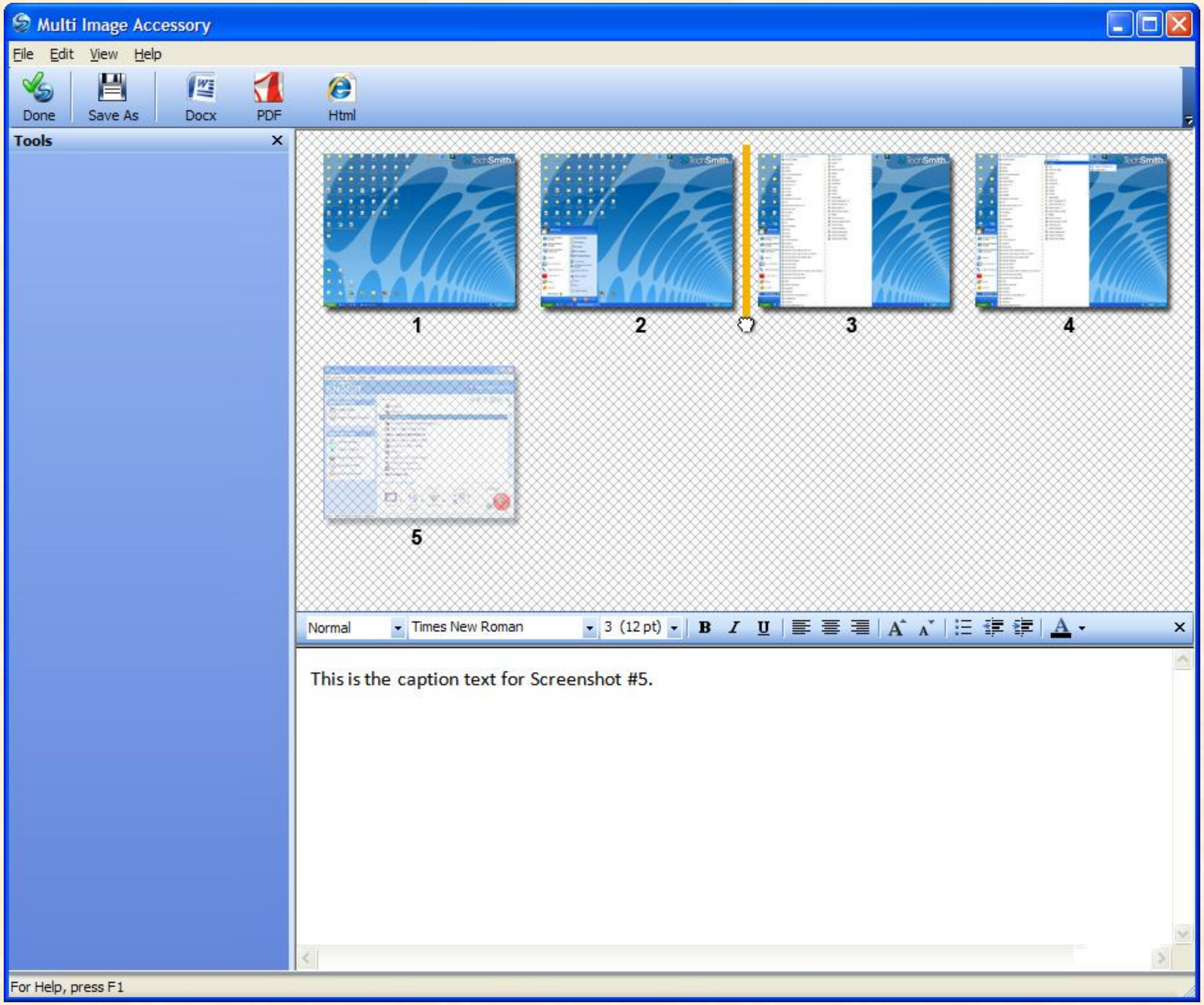


# Architecture Illustrated
# Mock Up 1 – Main Workspace



This is the caption text for Screenshot #1.

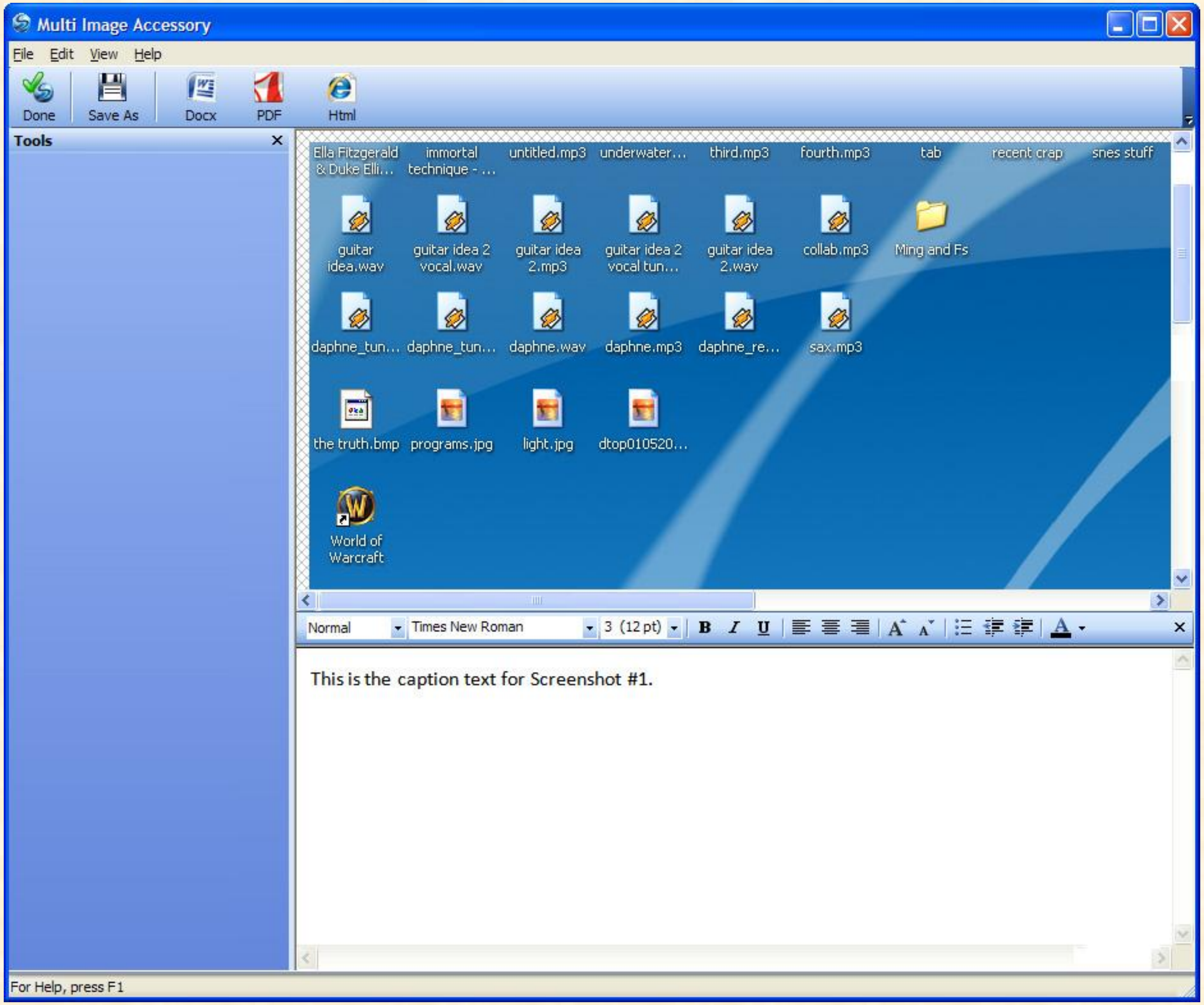
# Architecture Illustrated
# Mock Up 2 – Rearrange Images



This is the caption text for Screenshot #5.

Team 11: TechSmith

# Architecture Illustrated
# Mock Up 3 – View Full Image



This is the caption text for Screenshot #1.

Team 11: TechSmith

# Architecture Risks

- Each output type has its own different file format and some prove to be more difficult and complicated than others.
- The plug-in needs to maintain the look and feel of SnagIt, so the user does not know they are inside a separate application.
- The application needs to communicate properly with SnagIt using COM, which none of us have prior experience with.

# Project Schedule

1. **February 19th –** Prototype Due
   - **February 2nd –** Solidify all class diagrams and function prototypes.  Confirm with client that everything looks correct.
   - **February 9th –** Create a GUI layout and confirm with client that GUI appears as they perceived.  After layout has been created begin programmatically creating GUI.
   - **February 16th –** Begin implementing basic functionality.  Should have SnagIt sending a screen capture to MIA and MIA should be able to display that image.  Also try and get a temporary output working. (i.e. outputting a screen capture to word)

# Project Schedule

## **2. March 19th –** Prototype II Due

- **February 23rd –** Examine what revisions should be made to first version of prototype and prioritize which should be done first (Risk Analysis)
- **March 2nd –** Completion of one output format should be done.  This entails MIA accepting a screen capture from SnagIt and then being able to output to the designated output format.
- **March 9th –** GUI should be 100% finalized and set in stone.
- **March 19th –** Testing / Debugging all completed for Prototype II and should be a working solid prototype with *atleast* one working output format.

# Project Schedule

3. **April 9th –** Progress Report and Demo
   - Give a demonstration of our product and an updated progress report to class / client.
4. **April 23rd –** Team Videos
   - Complete video of our project from an end-user's perspective and also give a presentation on our project state.