# CS3733-D01: Software Engineering

## The KnowledgeBase Project : Component Implementation (Group)

**Interface Proposal Due:** Monday, April 9, in class
**Implementation Due:** Tuesday, April 24, 11:59pm (via electronic submission)
**Code Review:** Scheduled, between April 24 and the end of the term

---

# 1 Project Description

KB3733 is a technical support knowledge base. It supports customers searching for help on their own and technical support staff augmenting the troubleshooting information contained in the knowledge base. KB3733 uses a client/server architecture with a model-view-controller organization. In its full form, the system consists of a single server supporting simultaneous connections from multiple clients who wish to browse and access the knowledge base. In this form, communication between clients and the server is via TCP/IP. In a restricted form (say for small home offices), the system has only a single client that talks to the server directly, without a network involved.

The class as a whole will implement the KB3733 system, with each group implementing one component. Within the constraints of the agreed-on component interfaces, each group may implement their component as the members see fit. Different groups may use different data structures, or provide different levels of fault-tolerance, data integrity mechanisms, or scheduling algorithms, for example. After all components are turned in, we will create various versions of the knowledge base with different features, depending upon what each group implemented. Be creative!

## 1.1 Goals of the Project

This projet is designed to give you experience with several issues in software engineering:

- designing subsystem interfaces from requirements specifications,
- implementing subsystems in isolation from the rest of the system, relative to interface specifications,
- testing subsystems in isolation from the remainder of the system in which they will be used, and
- defending your design decisions to a code review team.

## 1.2 KB3733 Description

KB3733 provides a collection of troubleshooting guides for particular problems with products for a single company. Customers may search the guides against keywords or may do a more guided search based on the problem they are experiencing. In the latter case, the knowledge base will prompt the Customer with a series of instructions and questions, finally suggesting a solution for the Customer's problem. The solution may direct the Customer to download updated drivers or documentation, or may suggest that the Customer call technical support if the troubleshooting guide does not find a solution for the Customer's problem. Technical support staff can use the knowledge base in a similar way when talking to customers on the phone. In addition, technical support staff may add or update information in the knowledge base.

Specifically, KB3733 supports the following operations:

- Browse troubleshooting guide for a problem in step-by-step fashion
- Search the knowledge base by keyword
- Add a new product to the knowledge base (tech only)
- Add a new problem to the knowledge base (tech only)
- Annotate any step in the trouble shooting guide with notes or suggestions for technical support staff (tech only)

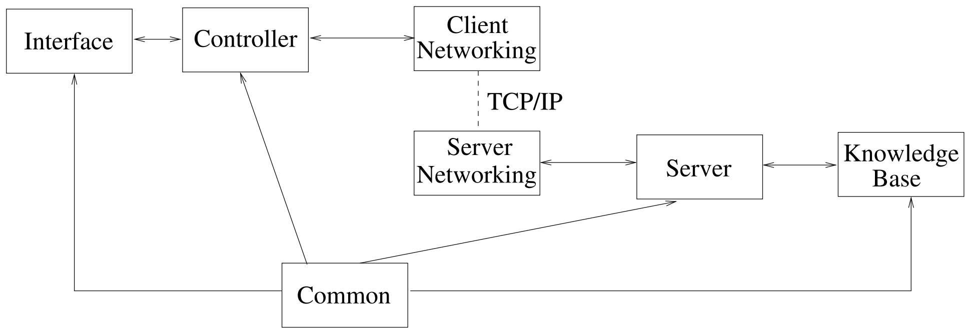


Figure 1: The KB3733 subsystem architecture.

- Add downloads to any instruction or solution in the knowledge base (tech only)
- Insert new instructions, questions, or solutions into the knowledge base (tech only)
- Modify existing instructions, questions, or solutions (tech only)
- Add a reference to a bug fix to every step in the troubleshooting guide for a particular problem (tech only)

Section 7.1 provides a set of use cases describing these operations in more detail.

# 2 Overall Architecture

Figure 1 shows the architecture for KB3733. The course staff will provide the KBCommon Subsystem as needed. Each group will be assigned (on the groups page) one of the following five portions of KB3733 to implement:

- the Interface: responsible for all displays between the users and the rest of the system.
- the Controllers: responsible for issuing commands to the knowledge base (via the server) to carry out the user's requests.
- the Networking System (both Client Network and Server Network): responsible for communicating messages and data between the controllers and the server.
- the Server: manages requests for data from the knowledge base from the (multiple) controllers.
- the Knowledge Base: the repository of problems and troubleshooting information.

# 3 Project Logistics and What to Turn In

This project has three distinct phases: design, implementation, and defense. Each has its own due date and expectations as outlined in the rest of this section.

## 3.1 For the Design Phase (due Monday, April 9)

Starting from the use cases given in Section 7.1, propose imports and exports for your assigned subsystem. Your imports and exports are lists of classes and functions that form the communication boundary between your subsystem and the neighboring subsystems.

- For each class in your imports/exports, document the class interfaces that the class must or will satisfy.
- For each function in your imports/exports, document the types of its parameters, the type of its return value, and the exceptions that the function could raise.

Turn in this document electronically before class (ascii is fine) and **bring five copies of your proposal to class on April 9th**.

#### 3.1.1 Notes on the Design Phase

- Do not document classes or functions that are internal to your subsystem. Document only the classes and functions that you think should be in the imports and exports.
- Do not submit any of the UML diagrams or documentation of analyses that you develop to determine your interfaces (though we suggest that you produce some of this documentation and use it to implement the system!).
- You may find it useful to prototype your component while designing your interfaces. This effort will not be wasted, as you will likely be able to reuse much of your prototyping code once the class settles on the actual interfaces for the project (perhaps with some renaming of functions).
- In class on April 9, we will consider the various proposals and choose the actual interfaces to implement for the project.

### 3.2 For the Implementation Deadline (due Tuesday, April 24)

Using the turnin program, submit the following files:

- A brief advertisement (ascii) summarizing the special features of your component. We will use these to choose components to glue together during the knowledge base integration day (during the week after the components have been turned in). Use the filename *advert.txt*.
- A single file for each subsystem that your group is required to implement. **The final expression in each file should evaluate to the unit implementing your required component** (this makes the component names irrelevant across projects). Name your files according to the following table;

| Component | File Name |
|---|---|
| Client Interface | interface.ss |
| Client Controller | controller.ss |
| Server | server.ss |
| Networking | clientnetwork.ss, servernetwork.ss |
| Knowledge Base | kbase.ss |

- A time log showing the time that your group spent in each activity of the project (understanding interfaces, implementing, testing, and documenting your component(s)). I suggest keeping a daily log so that you don't forget to compile this data.

### 3.3 For the Code Review (after submitting the implementation)

After the components are due, each group will do a 30 minute code review with the course staff. In this review, we will look at your code and ask you to justify your design and implementation decisions. We will also look at issues including (but not limited to) the cleanliness of your code, uses of helper functions, whether you reused code where possible, and whether you used classes in appropriate places. More details on the code walks are available on the web page.

## 4 Miscellaneous Notes

- As you think about your subsystem interfaces, keep our subsystem design goals in mind: we want subsystems that can be (a) easily exchanged within in system and (b) reused in other projects whenever possible. If your subsystem is something that we could conceivably use in another product, your interfaces should reflect the decoupling between your subsystem and the specifics of KB3733.

- MzScheme provides many helpful packages, such as hash tables, semaphores, and sorting routines. Always check the documentation or ask on the discussion board before investing effort in implementing common data structures and/or algorithms. In addition, MzScheme has good GUI and CGI libraries that could be helpful for the interface teams.
- Consider potential error cases carefully. Consider ways to maintain the integrity of the data that your component manages and to provide fair and protected access to services that your component provides (*i.e.*: the server should prevent one user from dominating access to the knowledge base). Every component should provide basic features along these lines. Your component advertisement should mention the special features that you implement.
- CVS, the version control system, may help you manage versions of and multi-programmer edits to, your project files. There is a pointer to CVS on the web page. If you have questions about or need help setting up CVS, ask (asking on the discussion board for this would be wise, as the course staff mostly works on the CS network rather than the CCC network).
- General rule: develop and test your code outside of units, then add the unit and signature structure at the end. Code is much easier to test outside of a unit, where all of the helper functions are visible.

# 5 Grading Criteria

We will review your interface proposals to grade the design phase of the project. We will be looking mainly for completeness: to what extent did you capture your subsystem's interactions with the rest of the the system? We will also check that you supplied the information required for each import/export as described in Section 3.1. Your grade on the component implementation will be assigned entirely based on your code review. If you can't defend your design decisions and your code structure, you will receive very little credit for the project (in other words, saying "it works" won't carry much weight).

# 6 Collaboration Policy

You are free to discuss this project across groups. Cross-group discussions with groups implementing neighboring subsystems may be particularly helpful during the design phase. However, all work turned in for this project (interface proposal and implementation) must be the work of the individual group members. **Do not share your code with other groups.**

# 7 The Use Cases and Glossary

Design your subsystem interfaces with respect to the use cases provided in this section. Do not add operations that are not represented in these use cases. While these use cases leave out several aspects of a full knowledge base system, this subset will keep the project size manageable.

## 7.1 Use Cases

```
======================================================================

UseCaseName     StartCustomerClient
                (includes BrowseBase, SearchBase)

Part. Actor     Initiated by Customer

Entry Condition 1. Customer starts the CustomerClient program

Flow of Events  2. Customer gets a screen with the following options:
```

```
                    - browse the KnowledgeBase
                    - search the KnowledgeBase.

Exit Condition  3. Customer selects an option.  If the Customer
                   selects the Browse option, go to the BrowseBase use
                   case.  If the Customer selects the Search option,
                   go to the SearchBase use case.

=====================================================================

=====================================================================

UseCaseName     StartTechClient
                (includes BrowseEditBase, SearchBase, AddProduct,
                          DisplayHelpDesk, MarkProblemSolved)

Part. Actor     Initiated by Tech

Entry Condition 1. Tech starts the TechClient program

Flow of Events  2. Tech gets a screen with the following options:
                    - browse/edit the KnowledgeBase
                    - search the KnowledgeBase
                    - add a product to the KnowledgeBase
                    - view the HelpDesk for a problem
                    - flag a problem with a bug fix

Exit Condition  3. Tech selects an option.  If the Tech selects the
                   Browse option, go to the BrowseEditBase use
                   case.  If the Tech selects the Search option, go to
                   the SearchBase use case. If the Tech selects the
                   AddProduct option, go to the AddProduct use case.
                   If the Tech selects the ViewHelpDesk option, go to
                   the DisplayHelpDesk use case.  If the Tech selects
                   the FlagFix option, go to the MarkProblemSolved use
                   case.

=====================================================================
=====================================================================

UseCaseName     BrowseBase

Part. Actor     Initiated by User
                Communicates with KnowledgeBase

Entry Condition 1. User selects a BrowseBase option

Flow of Events  2. System displays the products for which help is
                   available and prompts the User to select a
                   product. User selects a product.

                3. System contacts KnowledgeBase for the set of
                   problems for which help is available for the
                   selected product.  System displays problems to the
                   User.  User selects a problem.

                4. System displays the first HelpItem in the problem's
                   HelpDesk.  User sees the following options
```

```
                   (depending on the nature of the current HelpItem):
                     - ContinueFromInstruction : If the current
                       HelpItem is an instruction, view the next
                       HelpItem in the helpdesk.  Repeat this step.
                     - SelectQuestionAnswer : If the current HelpItem
                       is a question, view the HelpItem associated
                       with the selected answer.  Repeat this step.
                     - StartOver : returns User to the initial screen
                       that is appropriate for that User (the
                       StartCustomerClient screen for Customers, the
                       StartTechClient screen for Techs).
                   If the current HelpItem is a solution, User sees
                   the solution text, all downloads that are available
                   for that solution, and a StartOver option that
                   returns User to the appropriate initial screen.

Exit Condition  5. User returns to the initial screen.

=====================================================================
=====================================================================

UseCaseName     SearchBase

Part. Actor     Initiated by User
                Communicates with KnowledgeBase

Entry Condition 1. User selects a SearchBase option.

Flow of Events  2. System presents a SearchForm to the User.
                   User fills in form, supplying keywords to look for
                   in the search.  User submits form using a
                   StartSearch option.

                3. System creates SearchData from the User's keywords
                   and sends it to the KnowledgeBase.  KnowledgeBase
                   returns the text of all HelpItems that contain all
                   of the given keywords in a SearchResult.

                4. System displays the text in the SearchResult to the
                   User, with an option for the User to see any one of
                   the HelpItems in detail, as well as a StartOver
                   option.  If User selects an item, System displays
                   the selected item with the information as indicated
                   in the BrowseBase use case.  If User selects
                   StartOver, System displays the appropriate initial
                   screen for the User.

Exit Condition  5. System displays either the initial screen or the
                   screen for a particular item to the User.

=====================================================================
=====================================================================

UseCaseName     AddProduct
                (includes AddProblem)

Part. Actor     Initiated by Tech
                Communicates with KnowledgeBase
```

```
Entry Condition 1. Tech selects an AddProduct option.

Flow of Events  2. System displays an AddProductForm to the Tech.
                   Tech fills in the name of the new product and
                   submits the form by selecting a CreateProduct
                   option.

Exit Condition  3. System goes to the AddProblem use case.

=====================================================================
=====================================================================

UseCaseName     BrowseEditBase
                (extends BrowseBase step 4 with details specific to
                 what Techs can do to edit the KnowledgeBase)
                (includes UpdateHelpItem, InsertHelpItem,
                 AnnotateHelpItem)

Flow of Events  4. At each HelpItem, Techs see the following
                   additional options:
                   - UpdateHelpItem : go to the UpdateHelpItem use caes
                   - InsertHelpItem : go to the InsertHelpItem use
                     case; this option is not available on Solution
                     items.
                   - AnnotateHelpItem : go to the AnnotateHelpItem use
                     case
                   Techs also see all annotations on each HelpItem
                   while browsing.

=====================================================================
=====================================================================

UseCaseName     DisplayHelpDesk

Part. Actor     Initiated by Tech
                Communicates with KnowledgeBase

Entry Condition 1. Tech selects an ViewHelpDesk option,

Flow of Events  2. System displays the products and prompts the Tech
                   to select a product.  Tech selects a product.

                3. System contacts KnowledgeBase for the set of
                   problems available for the selected product.
                   System displays problems to the Tech.  Tech selects
                   a problem.

                4. System asks KnowledgeBase for a summary of the help
                   desk contents for that problem.  KnowledgeBase
                   creates a HelpDeskSummary and sends it to the
                   System.

Exit Condition  5. System displays the summary to the Tech, with
                   options for the Tech to select any HelpItem in the
                   summary.

=====================================================================
```

```
=====================================================================

UseCaseName     ProductNameExists
                (extends AddProduct at the end of Step 2)

Entry Condition 1. Name provided in a NewProductInfo is the name of an
                   existing product in the KnowledgeBase.

Flow of Events  2. System informs Tech that selected name is in use
                   and redisplays the AddProductForm to request a new
                   product name.

Exit Condition 3. Use case proceeds as in Step 2 of AddProduct.

=====================================================================
=====================================================================

UseCaseName     AddProblem

Part. Actor     Initiated by Tech
                Communicates with KnowledgeBase

Entry Condition 1. Tech selects an AddProblem option for a particular
                   product

Flow of Events  2. System displays the products in the KnowledgeBase
                   and prompts the Tech to select a product. Tech
                   selects a product.

                3. System presents a NewProblemForm to the Tech.  Tech
                   fills in a description of the new problem and
                   sumbits the form to the System using a
                   AddNewProblem option.

                4. System creates a NewProblemInfo containing the
                   description and sends it to the KnowledgeBase.
                   KnowledgeBase creates a new HelpDesk for the
                   problem and sets the first HelpItem in the problem
                   to be the default solution.  KnowledgeBase sends an
                   acknowledgement to the System that the Problem has
                   been added.

Exit Condition  5. System receives the acknowledgement and displays
                   the collection of problems avaiable for the product
                   (including the new problem) to the Tech.

=====================================================================
=====================================================================

UseCaseName     UpdateHelpItem

Part. Actor     Initiated by Tech
                Communicates with KnowledgeBase

Entry Condition 1. Tech selects an UpdateHelpItem option.

Flow of Events  2. System presents one of the following displays
                   depending upon the nature of the selected
```

```
                     HelpItem.  System also displays an UpdateText
                     option.
                     - If HelpItem is an instruction, system displays
                       the current text of the instruction for editing.
                     - If HelpItem is a question, system displays the
                       current text of the question for editing.
                     - If HelpItem is a solution or an instruction,
                       system displays the current text of the solution,
                       as well as an AddDownload option to augment the
                       downloads available for the solution.

                  3. If Tech selects an UpdateText option, system
                     creates an UpdateItem form containing the new text
                     and sends it to the KnowledgeBase.  KnowledgeBase
                     performs the update and sends and acknowledgement
                     to the System.

Exit Condition    4. Tech sees the revised display for the HelpItem

=====================================================================
=====================================================================

UseCaseName       InsertHelpItem

Part. Actor       Initiated by Tech
                  Communicates with KnowledgeBase

Entry Condition 1. Tech selects an InsertHelpItem option from an
                     instruction or question HelpItem display.

Flow of Events    2. System displays AddInstruction, AddQuestion, and
                     AddSolution options to the Tech.  If the current
                     HelpItem is a question, System also displays a
                     AddAnswer option.  Tech selects an option.

                  3. - If the Tech selects the AddInstruction option,
                       System prompts Tech for the text of the new
                       instruction, creates a NewInstructionInfo from
                       the response, and sends that form to the
                       KnowledgeBase to insert the new instruction after
                       the current HelpItem.

                     - If the Tech selects the AddQuestion option,
                       System prompts Tech for the text of the new
                       question and a list of answers.  System creates a
                       NewQuestionInfo from the response and sends
                       that form to the KnowledgeBase to insert the new
                       question list as the FollowUp item for the
                       current HelpItem.

                     - If the Tech selects the AddAnswer option, System
                       prompts Tech for the text of the new Answer.
                       System creates a NewAnswerInfo from the response
                       and sends that form to the KnowledgeBase.
                       KnowledgeBase inserts the new answer into the list
                       of answers for the question, and sets the default
                       solution as the FollowUp item for the current
                       HelpItem.
```

```
                   - If the Tech selects the AddSolution option,
                     System goes to the NewSolution use case to have
                     the Tech create the new solution.  System tells
                     KnowledgeBase to create the new solution and to
                     make it the FollowUp item to the current HelpItem.

                   When the update is complete, KnowledgeBase sends an
                   acknowledgement to the System.

Exit Condition  4. Tech receives acknowledgement from System that the
                   insertion has been completed and System redisplays
                   the HelpItem to the Tech.

=====================================================================
=====================================================================

UseCaseName     HelpItemLocked
                (extends UpdateHelpItem in the case that another Tech
                 is editing the same HelpItem)

Entry Condition 1. Tech selects an UpdateHelpItem option for a
                   HelpItem that another Tech is currently editing.

Flow of Events  2. System informs Tech that HelpItem is currently
                   being updated and returns Tech to display for the
                   selected HelpItem.  When the KnowledgeBase has
                   finished updating the HelpItem, System displays the
                   revised HelpItem for the Tech if the Tech is still
                   viewing that HelpItem.

Exit Condition  3. Tech sees revised display or moves to another
                   display.

Special Reqts   Techs cannot be prevented from making updates
                indefinitely.  If a Tech keeps trying to update a
                HelpItem, he is eventually given access to do so.

=====================================================================
=====================================================================

UseCaseName     AnnotateHelpItem

Part. Actor     Initiated by Tech
                Communicates with KnowledgeBase

Entry Condition 1. Tech selects an AnnotateItem option for a
                   particular HelpItem.

Flow of Events  2. System prompts Tech for the text of the new
                   annotation.  Tech provides text in a
                   NewAnnotationForm, then submits that form to the
                   System using a AddAnnotation option.  System
                   creates a NewAnnotationInfo from the form and sends
                   it to the KnowledgeBase.

                3. KnowledgeBase adds the new annotation to the
                   existing annotations for the HelpItem and sends an
```

```
                  acknowledgement to the System.

Exit Condition  4. System displays the updated HelpItem to the Tech.

Special Reqts   After an update, the display of any other Tech who is
                viewing the HelpItem should be updated.

=====================================================================
=====================================================================

UseCaseName     AddDownload

Part. Actor     Initiated by Tech
                Communicates with KnowledgeBase

Entry Condition 1. Tech selects an AddDownload option for a particular
                   instruction or solution item.

Flow of Events  2. System displays a NewDownloadForm to the Tech.
                   Tech fills in a description of the download, the
                   path to the file to be downloaded, and indicates
                   whether it is a driver or a document.  Tech submits
                   the form using a SubmitDownload option.  System
                   creates a NewDownloadInfo from the submitted
                   information and sends it to the KnowledgeBase.

                3. KnowledgeBase updates the downloads available for
                   the HelpItem and sends an acknowledgement to the
                   System.

Exit Condition  4. System refreshes Tech's display of the HelpItem to
                   show the new download.

=====================================================================
=====================================================================

UseCaseName     MarkProblemSolved
                (includes CreateSolution)

Part. Actor     Initiated by Tech
                Communicates with KnowledgeBase

Entry Condition 1. Tech selects the MarkProblemSolved option for a
                   particular problem

Flow of Events  2. System prompts Tech for a new solution or a
                   reference to an existing solution.  If Tech chooses
                   the NewSolution option, go to the CreateSolution use
                   case.  System sends a ProblemSolvedInfo to the
                   KnowledgeBase that indicates the problem and the
                   corresponding solution.

                3. KnowledgeBase augments each HelpItem in the problem
                   with a BugFixFlag that refers to the submitted
                   solution, then sends an acknowledgement to the System.

Exit Condition  4. System displays the initial tech screen to the Tech.
```

```
======================================================================
======================================================================

UseCaseName     CreateSolution

Part. Actor     Initiated by Tech
                Communicates with KnowledgeBase

Entry Condition 1. Tech selects a CreateSolution option

Flow of Events  2. System displays a NewSolutionForm.  Tech fills in
                   the text of the solution and the paths to any
                   downloads that should be available with the
                   solution.  Tech submits the form to the system
                   through a SubmitNewSolution option.

                3. System creates a NewSolutionInfo and sends it to
                   the KnowledgeBase.  The KnowledgeBase creates a
                   HelpItem for the solution, assigns it a reference
                   number, and returns that reference number to the
                   System.

Exit Condition  4. The System displays a message containing the new
                   reference number to the Tech.

======================================================================
======================================================================

UseCaseName     NoMatchesFound
                (extends SearchBase in step 3 if the KnowledgeBase
                 finds no matches for the given keywords)

Entry Condition 1. KnowledgeBase finds no HelpItems that contain all
                   of the keywords provided for a search.

Flow of Events  2. KnowledgeBase sends a NoMatchesFoundMsg to the
                   System.

Exit Condition  3. System displays the message to the User and
                   displays a SearchForm with the User’s original
                   keywords filled in.  User can edit the form and
                   resubmit or can select the StartOver option to
                   return to the initial screen.

======================================================================
```

## 7.2 Glossary

- **Bug-fix Flag**: A special annotation on a HelpItem to indicate that the problem has been solved via a bug fix to the drivers. Help Items with bug-fix flags contain references to the new drivers. Help Items with Bug-fix flags remain in the KnowledgeBase to support Customers who have older hardware that requires the old drivers.
- **Customer**: someone who is not a Tech, yet wants to use the KnowledgeBase to troubleshoot a problem with a product.
- **Default Solution**: A solution that instructs the reader to contact technical support because no further help is currently available for the given problem. New problems are assigned the default solution while Techs edit the KnowledgeBase with a more detailed HelpDesk for the problem. May also remain the actual solution in certain cases.
- **FollowUp Item**: The next HelpItem to display from a given HelpItem in a HelpDesk. Instructions and Answers have Followup items; Solutions do not.
- **HelpDesk**: a series of instructions and questions that suggest solutions for a particular problem.
- **HelpItem**: an instruction, response to a question, or solution in a HelpDesk. Instructions tell Users to perform a specific action. Questions provide Users several possible responses, one of which should fit their situation. Solutions give final instructions for solving a problem.
- **KnowledgeBase**: a database of information related to problems with products that the company manufactures. Contains step-by-step troubleshooting guides (HelpDesks), downloads for documentation and drivers, and notes made by Techs on the various troubleshooting hints.
- **Problem**: a specific scenario under which a particular product is not working properly.
- **Tech**: An employee of technical support. Authorized to update entries in the KnowledgeBase.
- **User**: either a Tech or a Customer.