

E BOARD

DEVELOPER DOCUMENTATION

Version – 2.0

Date– 2016/05/22

Team –ELECTRO BLITZ

PROJECT MEMBERS

^{#1} ROBIN PHILIP JOSEPH, ^{#2} NAGA SHRUTI ADIDAMU, ^{#3} SHANMUKHA SAI
BHEEMISETTY, ^{#4} SHIVA SAI SUNKARI, ^{#5} RAJASHEKAR REDDY RAMIDI, ^{#6}
VIGNESH KUNA, ^{#7} DIVYA KONDAVEETI, ^{#8} DHARANI NIMMAGADDA, ^{#9} B V S
PREETHIKA POTLURI, ^{#10} SINDHU VASIREDDY, ^{#11} ROHITH REDDY
JONNALAGADDA

Students of Telecommunication Systems, Blekinge Institute of Technology SE-371 79 Karlskrona,
Sweden.

1. PREFACE:

The main aim of the project is to develop a basic Internet Whiteboard, which facilitates communication between employee of the consultancy (ConTech) and its customers. It helps them to work together, despite being far away. This is revised version of the Design document (version -2.0).

In the remainder of the document, all the modules of the internet whiteboard are described in detail and the test plan of each module has been made.

1.1 LOG MODIFICATIONS:

Release version 2.0 on 2016/05/22

- Reload (on click), reload (automatic), timestamp and inserter functions are added in tool management (see section 3.1.12,3.1.13,3.1.14,3.1.15)
- locker, unlocker, modselection, send mail, mod_ok functions are added (see section 3.2.7,3.2.8,3.2.10,3.2.11,3.2.9)
- Added threading functionality(section 3.4), my thread (section 3.4.1), reg_thread (section 3.4.2), retrieve (section 3.4.3)
- Added ChangelP (section 3.5.1), Setup (section 3.5.2), Ok(section 3.5.3), show1(section 3.5.4), show2(section 3.5.5), show3(section 3.5.6), toSSLornottoSSL (section 3.5.7), on_closing (section 3.5.8), checkmod(section 3.5.9), start_session(section 3.5.10) functions for main program.
- Added UPDATE (section 4.3) function to the databaser.

Release version 1.0 on 2016/05/15

- Initial release

2. GLOSSARY AND ABBREVIATIONS

ConTech	Consulting firm (Costumer)
DB	Database
GUI	Graphical User Interface
IP	Internet Protocol
MySQL	Open-Source relational database management system
Python	Programming Language
PDF	Portable document format
SHA1	Cryptographic Hash Function
Timestamp	Indicates Date and Time in a standard format

3. EBOARD FUNCTIONALITY

The entire functionality of the product is divided into five parts, four classes and one main code:

1. Tool Management
2. Sheet Management
3. User Management
4. Threading
5. Main code

3.1 TOOL MANAGEMENT:

3.1.1. `__init__(self)`:

It is a constructor of the class, Toolmgmt () which is used to initialize the below variables:

`self.can` = canvas object

`self.canvas_name` = string which stores name of current canvas.

3.1.2. `Clear(self)`:

The above function is used to clear the whiteboard. After clearing the whiteboard tools are set to default values like thickness, color.

3.1.3. `Pencil (self)`:

The above function is used to implement the pencil functionality in the whiteboard.

`getClick(self, event)` : gets the co-ordinates of the mouse position on left click.

`pencilDraw(self, event)` : drawing functionality is executed through motion of mouse.

3.1.4. `Eraser (self)`:

This function is used to erase the sketch created.

`getClick(self, event)` : gets the co-ordinates of the mouse position on left click.

`eraseStuff(self,event)`: erasing functionality is executed through motion of mouse.

3.1.12 Additional tool functions:

`font1 (self)`:

`font2 (self)`:

`font3 (self)`:

`font4 (self)`:

`font5 (self)`:

`font6 (self)`:

The above functions are used to change the font style for the text being inserted.

Size1(self):
size2(self):
size3(self):
size4(self):
size5(self):
size6(self):

The above functions are used to change the size of the text being inserted in the Whiteboard.

Bold(self):
Normal(self):
Italic(self):
Roman(self):

The above functions are used to change the style of the text being inserted in the Whiteboard.

Redclr(self):
Greenclr(self):
Blueclr(self):
Blackclr(self):

The above functions are used to change the outline color of pencil, line, polyline, oval, rectangle, arrow tools.

Redclrf(self):
Greenclrf(self):
Blueclrf(self):
Blackclrf(self):

The above functions are used to select the color to be filled in the rectangle/square and oval/circle.

thck1(self):
thck3(self):
thck5(self):

The above functions are used to change the thickness of pencil, line, polyline, oval, rectangle, arrow tools.

3.1.5.Oval(self):

This function is used to draw an oval in the Whiteboard.

ovalStart(self, event) : gets the co-ordinates of the mouse position on left click.

ovalEnd(self,event): Stores the command to draw the oval ,which is inserted into the database as a string on the release of mouse click

3.1.6.Line(self):

This function is used to draw a line.

getClick(self, event) : gets the co-ordinates of the cursor position on left mouse click.

drawLine(self, event) : Stores the command to draw the line ,which is inserted into the database as a string on the release of mouse click

3.1.7.Rect(self):

This function is used to draw a rectangle.

rectStart(self,event): gets the co-ordinates of the cursor position on left mouse click.

rectEnd(self,event): The command to draw the rectangle is inserted into the database as a string on the release of mouse click.

3.1.8.Arrow(self):

This function is used to draw an arrow on the Whiteboard.

getClick(self, event) : gets the co-ordinates of the cursor position on left mouse click.

arrowdraw(self, event) : Stores the command to draw the arrow ,which is inserted into the database as a string on the release of mouse click.

3.1.9.Polyline(self):

This function is used to create polyline upon the mouse clicks.

getClick(self, event) : gets the co-ordinates of the cursor position on left mouse click.

drawpolyline(self, event): stores the command to draw a line,which inserted into the database as a string and to reset the co-ordinates of the cursor to the new value.

polylineend(self, event): ends the functionality of drawpolyline.

3.1.10.Text_input(self):

This function is used to insert text on the Whiteboard.

Text(self, event) : a dialog box to enter text appears on left mouse click and options to change style of text are obtained.

Upon the right mouse click on the whiteboard, all the values of the **Text_input(self)** function are set to the default values through the execution of **textend(self,event)** function.

3.1.11.Undo(self):

To obtain undo functionality the last modification on the whiteboard is retrieved from the database and deleted from the whiteboard

3.1.12 reload_click(self):

To reload all the modifications in the database by a mouse click.

get_modifications(self):Retrieves all the modifications in the database as a list.

reload_event(self):binds the mouse click to reload_click function.

3.1.13 reload_automatic(self):

To reload all the modifications in the database from the beginning automatically with 1 sec delay between each modification.

3.1.14 Timestamp(self):

To generate a timestamp whenever the function is called.

3.1.15 inserter(self,string):

To store the modification index, sheet number, modifications , user responsible for the modification and time stamp into the database.

3.2.USER MANAGEMENT:

3.2.1.Login(self):

Upon clicking the login button, the user is directed to another frame which consists of the username and password.

In this frame, the input for username and password are accepted from the user.

3.2.2.Register (self):

Upon clicking the register button, the user is directed to another frame which consists of username and Email ID.

In this frame, the input for username and Email ID are accepted from the user and the login details are E-mailed to the user.

3.2.3.Bac(self):

This function is used to direct the user back to the login page.

3.2.4.Ok_log(self):

This function is used to update the user status in the database and to compare the user details in case of existing user.

3.2.5.Ok_reg(self):

This function is used to generate a random password and to insert the new user details into the database.

3.2.6.Logout(self):

This function is used to remove all the frames and displays the first frame. The status of the user is also updated as logged out in the database.

3.2.7.locker(self):

To lock the users in the session from making any modifications on the whiteboard.

checkLock(self):

If the moderator status is "TRUE" user can lock the whiteboard.

3.2.8.unlocker(self):

To unlock the users to make modifications on whiteboard.

checkLock(self):

If the moderator status is "TRUE" user can unlock the whiteboard.

3.2.9.mod_ok(self):

To push the status of the moderator into the database.

3.2.10.modselection(self):

To select the moderator from the list of users with radio buttons.

3.2.11.sendMail(self,toaddr, regus, regpass):

smtplib module is imported and also fromaddr is given ,forwarding the mail consisting of username and password to the entered mail address.

3.3. SHEET MANAGEMENT:

3.3.1.Show(self):

This function packs the particular selected sheet.

3.3.2.Hide(self):

This function is used to close the particular selected sheet.

3.4 THREADING:

3.4.1.myThread ():

It continuously checks the lock status of the user. It changes the value of lock_flag to TRUE if he is locked and to FALSE if he is not locked.

3.4.2.reg_thread ():

This thread is used to send a mail to the user, whenever he is registered either by Admin or Employee.

3.4.3.Ret ():

This thread is used to retrieve modification data from the database and replicate those modifications on the whiteboard sheet.

3.5.MAIN PROGRAM:

3.5.1.ChangeIP():

The Admin is provided with change IP button.This function provides the Admin with the option to change the IP address.

3.5.2.Setup():

To create the tables in the database if tables doesn't exist.

3.5.3.OK():

The Admin is provided with the OK button in the IP reconfigure frame.

Upon providing the new IP address and clicking this button, the IP address is changed and the user is directed to the previous frame with no change IP button.

3.5.4.show1():

This function displays the SHEET 1 and hides the rest of the sheets.

3.5.5.show2():

This function displays the SHEET 2 and hides the rest of the sheets.

3.5.6.show3():

This function displays the SHEET 3 and hides the rest of the sheets.

3.5.7.toSSLornottoSSL(thelink, checker):

If the checker value is "YES" the function tries to open the link with https and if the checker value is "NO" the function tries to open the link with http.

3.5.8.on_closing():

This function is executed when the user clicks the close button. This function is used to close all the frames and the status of the user is updated as logged out.

3.5.9.CheckMod(event):

If the moderator status is "TRUE" lock,unlock and undo buttons are displayed.

3.5.10.start_session():

This function enables the employee to start a new whiteboard session and designates him as the moderator.

3.5.11. Employee():

This function is used when a new employee is registered.

This function calls the **register(self)** function which directs the user to enter the username and Email address as input.

3.5.12.Customeremp():

This function is used when employee registers a new customer.

This function calls the **register(self)** function which directs the user to enter the username and Email address as input.

3.5.13.Customeradm():

This function is used when Admin registers a new customer.

This function calls the **register(self)** function which directs the user to enter the username and Email address as input.

A GUI window is created using the command “root = Tk()”

The GUI window is closed on clicking the close button using the command root.protocol("WM_DELETE_WINDOW", on_closing).

```
button0 = Button(downstairs, text="Clear", command = toolobj.Clear)
```

The above command is a sample to create the required buttons.

```
mainframe = Frame(root)
frame1= Frame(root)
frame2 = Frame(root)
frame3 = Frame(root)
frame4 = Frame(root)
frame5 = Frame(root)
```

The above commands are used to create frames on the Tkinter window.

```
canvas1 = Canvas(upstairs, bg = 'white')
canvas2 = Canvas(upstairs, bg = 'white')
canvas3 = Canvas(upstairs, bg = 'white')
```

These commands are used to create canvases with the background as white colour.

```
menu1 = Menu(mainframe)
```

This command creates a menubar in the Tkinter window.

```
Submenu = Menu(menu1)
menu1.add_cascade(label = "Colors", menu = Submenu)
Submenu.add_command(label="RED", command = toolobj.redclr)
Submenu.add_command(label="BLUE", command = toolobj.blueclr)
Submenu.add_command(label="GREEN", command = toolobj.greenclr)
Submenu.add_command(label="BLACK", command = toolobj.blackclr)
```

Above commands add a Colors submenu in the menubar, which has a drop down menu of four colors:

1. Red
2. Blue
3. Green
4. Black

When one of these colors is chosen, the tool we select will draw the shape in opted color.

```
editm = Menu (menu1)
menu1.add_cascade(label="Thickness", menu = editm)
editm.add_command(label="Thickness 1", command = toolobj.thck1)
editm.add_command(label="Thickness 3", command = toolobj.thck3)
editm.add_command(label="Thickness 5", command = toolobj.thck5)
```

Above commands add a Thickness submenu in the menubar, which has a drop down menu of three thicknesses:

1. Thickness 1
2. Thickness 2
3. Thickness 3

When one of these is selected, the thickness of the outline of the shapes will be changed to specific one.

```
thmen = Menu (menu1)
menu1.add_cascade(label="Fill Color", menu = thmen)
thmen.add_command(label="Red", command = toolobj.redclrf)
thmen.add_command(label="Blue", command = toolobj.blueclrf)
thmen.add_command(label="Green", command = toolobj.greenclrf)
thmen.add_command(label="Black", command = toolobj.blackclrf)
```

Above commands add a Fill Color submenu in the menubar, which has a drop down menu of four colors:

1. Red
2. Blue
3. Green
4. Black

When one of these is selected, rectangle/oval shapes will be filled by the chosen color.

```
sheetMenu = Menu (menu1)
menu1.add_cascade(label="Sheets", menu = sheetMenu)
sheetMenu.add_command(label="Sheet 1", command = show1)
sheetMenu.add_command(label="Sheet 2", command = show2)
sheetMenu.add_command(label="Sheet 3", command = show3)
```

Above commands add a Sheets submenu in the menu bar, which has a drop down menu of three sheets:

1. Sheet 1
2. Sheet 2
3. Sheet 3

This allows the moderator to change between sheets in a session.

```

fonts = Menu (menu1)
menu1.add_cascade(label="Text Font",menu=fonts)
fonts.add_command(label="Helvetica",command= toolobj.font1)
fonts.add_command(label="SimSun",command=toolobj.font2)
fonts.add_command(label="Times",command=toolobj.font3)
fonts.add_command(label="Stencil",command=toolobj.font4)
fonts.add_command(label="Magnetto",command=toolobj.font5)
fonts.add_command(label="French Script MT",command=toolobj.font6)

```

Above commands add a fonts submenu in the menu bar, which has a drop down menu of six fonts:

1. Helvetica
2. SimSun
3. Times
4. Stencil
5. Magnetto
6. French Script MT

This allows the user to type text in different fonts.

```

fontsize= Menu(menu1)
menu1.add_cascade(label="Text Font Size",menu=fontsize)
fontsize.add_command(label="16",command=toolobj.size1)
fontsize.add_command(label="20",command=toolobj.size2)
fontsize.add_command(label="24",command=toolobj.size3)
fontsize.add_command(label="28",command=toolobj.size4)
fontsize.add_command(label="32",command=toolobj.size5)
fontsize.add_command(label="36",command=toolobj.size6)

```

Above commands add a fontsize submenu in the menu bar, which has a drop down menu of six font sizes:

1. Size 16
2. Size 20
3. Size 24
4. Size 28
5. Size 32
6. Size 36

These allow the user to choose between different font sizes.

```

usmen = Menu (menu1)
menu1.add_cascade(label = "User", menu = usmen)
usmen.add_command(label="logout", command = userobject.logout)

```

This option allows the customer to logout of the current whiteboard session.

4.DATABASER FUNCTIONALITY

Import the following modules from python library:

MySQLdb and datetime

4.1 CLASS

Defined class : Databaser()

A class is defined with the name databaser.

4.2 INITIALIZATION:

The following parameters need to be initialized : self.DB_UserName ,self.DB_Password,self.DB_PORT
self.DB_DBName ,self.DB_Table, self.DB_SESSION_Table .

The initialization operation creates an empty object. Many classes like to create objects with instances customized to a specific initial state. Therefore a class may define a special method named `__init__()`, like this.

4.3 UPDATE ()

This function reads the IP from 'eboard.cnf' file and assigns it to DB_IP variable. This is necessary when Admin changes the server IP.

4.4 CREATE_TABLE ():

Creates a new user data table in the MySQL database to store the information related to the users in the session. Name of the table is 'user_table'.

4.5 CREATE_SESSION_TABLE ():

Creates a new table in the MySQL database to store the data related to modifications done on the sheet. Name of the table is 'Table_1'.

4.6 POP_USER ():

It returns a list of all existing users in the session.

4.7 POP_LOCK ():

It returns a list of all the users who are locked.

4.8 POP_USER_TYPE ():

It returns the type of a specific user.

4.9 POP_MODERATOR ():

It returns a list with the name of the moderator.

4.10 PUSH_DATA ():

It inserts the name and type of the user into the user data table maintained in MySQL database.

4.11 PUSH_LOCK ():

It sets the lock status of the specified user as YES.

4.12 PUSH_LOGGEDIN ():

It sets the LOGGEDIN status of the specified user as YES.

4.13 PUSH_LOGGEDOUT ():

It sets the LOGGEDIN status of the specified user as NO.

4.14 PUSH_UNLOCK ():

It sets the lock status of the specified user as NO.

4.15 PUSH_MODERATOR ():

It sets the moderator status of the specified user as YES.

4.16 kicker ():

It removes the specified user from the session upon expiry of validity period.

4.17 PUSH_UNMODERATOR ():

It sets the moderator status of the specified user as NO.

5.REFERENCES:

[1] ConTech product request, P5: Internet Whiteboard, 2016. [Online]. Available for itslearning users:
<https://bth.itslearning.com/ContentArea/ContentArea.aspx?LocationID=6504&LocationType=1>